HEWLETT-PACKARD

HP-16C

BENUTZERHANDBUCH



HINWEIS

Hewlett-Packard übernimmt weder ausdrücklich noch stillschweigend irgendwelche Haftung für die in diesem Handbuch dargestellten Programm-Materialien und Tastenfolgen – weder für ihre Funktionsfähigkeit noch für ihre Eignung für irgendeine spezielle Anwendung. Die Tastenfolgen und Programme dienen lediglich der beispielhaften Illustration; der Benutzer trägt das volle Risiko bezüglich ihrer Güte und Durchführbarkeit. Sollten sich Programm-Materialien und/oder Tastenfolgen als fehlerhaft erweisen, trägt der Benutzer (und nicht Hewlett-Packard oder irgendeine dritte Seite) die Kosten für die Korrekturen und alle Neben- und Folgeschäden. Hewlett-Packard haftet für keinerlei Neben- oder Folgeschäden im Zusammenhang mit oder als Folge aus der Lieferung, Benutzung oder Leistung der Programm-Materialien und Tastenfolgen.



HP-16C

Benutzerhandbuch

00016-90002

Printed in Canada

Einleitung

Der HP-16C ist ein vielseitiger und einzigartiger Rechner, speziell entwickelt für Informatiker und alle, die mit Computern und Mikroprozessoren arbeiten, sei es in der Entwicklung oder in der Programmierung. Er bietet die folgenden außergewöhnlichen Anwendungsmöglichkeiten:

- Ganzzahlige Arithmetik in vier Zahlensystemen (hexadezimal, dezimal, oktal und binär) in den Modi Einerkomplement, Zweierkomplement und Unsigned (ohne Vorzeichen).
- Variable, vom Anwender bestimmbare Wortlänge bis zu 64 Bit.
- Logische Operationen und Bitmanipulationen.
- 203 Bytes Benutzerspeicher, entsprechend bis zu 203 Programmzeilen.
- Gleitkomma-Arithmetik.
- Permanentspeicher für zeitlich unbegrenzte Daten- und Programmspeicherung.
- Extrem niedriger Stromverbrauch und lange Batterie-Lebensdauer.

Dieses Handbuch ist für den professionellen Anwender geschrieben, für jemanden, der mit den Prinzipien der Computerorganisation und mit Binäroperationen schon vertraut ist. Für einen schnellen Überblick und zum Nachschlagen der Rechnerfunktionen können Sie das Funktionsverzeichnis – die blauen Seiten vor dem Sachindex – benutzen.

Teil I des Handbuches, HP-16C Grundlagen, behandelt die speziellen Funktionen des HP-16C und die Logik der Umgekehrten Polnischen Notation. Teil II, HP-16C Programmierung, ist den Programmierungsmöglichkeiten des HP-16C gewidmet. Jeder Abschnitt in Teil II erklärt die entsprechenden Operationen zunächst allgemein, gibt dann Beispiele und behandelt schließlich detailliert spezielle Anwendungen. Dadurch wird es Ihnen erleichtert, sich einen Überblick über die Funktionsweise des Rechners zu verschaffen.

Die Abschnitte I und II beschreiben die Standardfunktionen von HP-Rechnern, während die Spezialfunktionen des HP-16C in den Abschnitten 3 bis 6 konzentriert sind.

Zuvor erhalten Sie im Abschnitt «Ein erster Überblick» auf den Seiten 10 bis 12 eine kurze Einführung in die Fähigkeiten des HP-16C.

In den Anhängen finden Sie Details zu den Fehler- und Flagbedingungen, Auflistungen von Operationen mit speziellen Charakteristika sowie Gewährleistungs- und Serviceinformationen. Der Funktionsindex am Ende des Handbuchs enthält Kurzbeschreibungen aller Tastenfunktionen mit Seitenangaben.

Inhalt

Der HP-16C: Ein erster Überblick

Der HP-16C ist ein vielseitiger Problemlöser, der sowohl im Integer- als auch im Gleitkomma-Modus arbeitet. Im Integer-Modus können Sie ganzzahlige Binärarithmetik, Umwandlungen der Zahlenbasis, Bitmanipulationen und logische Operationen ausführen. Im dezimalen Gleitkomma-Modus können Sie umfangreiche Gleitkomma-Berechnungen durchführen. Sie können ihn bei den Modi programmieren. Der Permanentspeicher des HP-16C speichert Daten und Programme unbegrenzt lange, bis er von Ihnen gelöscht wird.

Eine wichtige Eigenschaft des HP-16C ist sein extrem niedriger Stromverbrauch. Das macht ihn sehr leicht und handlich; ein Ladegerät wird überflüssig. Der Stromverbrauch ist so niedrig, daß eine Batterie bei normalem Gebrauch bis zu einem Jahr ausreicht. Auch werden Sie bei schwächer werdender Betriebsspannung lange vor dem Ausfall des Geräts gewarnt.

Weiterhin schaltet der HP-16C seine Anzeige automatisch aus, sobald für einige Zeit keine Eingabe mehr erfolgt ist, und schont damit seine Batterie.

Tastenfeldoperationen

Ihr HP-16C arbeitet mit Umgekehrter Polnischer Notation, einer Eingabelogik, die durch die [ENTER] Taste gekennzeichnet ist. Diese Logik macht Klammern überflüssig, ein Speicherstack übernimmt die Funktion der Klammerregister. Wir wollen dies anhand der arithmetischen Funktionen erläutern.

Wählen Sie bei eingeschaltetem Rechner (drücken Sie [ON], falls notwendig) eine Zahlenbasis (hexadezimal, dezimal, oktal oder dual), indem Sie eine der Tasten [HEX], [DEC], [OCT] oder [BIN] drücken. Die gewählte Basis wird durch die Symbole **h**, **o**, **d** oder **b** in der Anzeige gekennzeichnet. Voreinstellung (beim ersten Einschalten oder Löschen des Permanentspeichers) ist Hexadezimal. Die Spezifikation einer dieser Zahlenbasen schaltet den Rechner automatisch in den Integer-Modus. Mit [g][CLx] (einer blau gekennzeichneten Funktion) können Sie die Anzeige löschen (d.h. auf den Wert Null setzen).*

* Die meisten Tasten haben drei Funktionen. Um die in weiß *auf* die Taste aufgedruckte Primärfunktion auszuführen, brauchen Sie lediglich diese Taste zu drücken. Zur Ausführung der goldfarben oder blau gekennzeichneten Alternativfunktionen müssen Sie zuvor die goldfarbene Vorwahltaste [f] bzw. die blaue Vorwahltaste [g] drücken.

Zur Ausführung einer arithmetischen Operation sind zunächst die beiden Operanden, durch [ENTER] getrennt, und danach der Operator einzutasten. Die Funktion wird direkt nach Drücken der Taste ausgeführt, das Ergebnis erscheint sofort in der Anzeige. Wenn Sie eine falsche Ziffer eingetastet haben, können Sie den Fehler mit [BSP] löschen und danach die richtige Ziffer eingeben.

Berechnungen im Integer-Modus

Sobald Sie einen Zahlenbasis-Modus gesetzt haben, arbeitet der Rechner automatisch im Integer-Modus (d.h. mit einer ganzzahligen Arithmetik).

Zu berechnen*	Eingabe	Anzeige
(in Basis 2)	[g] [CLx] [BIN]	**0 b**
1111–1	1111 [ENTER] 1 [−]	**1110 b**
1111 × 11	1111 [ENTER] 11 [×]	**101101 b**

Sie können auch mit einem Wert rechnen, der sich schon in der Anzeige befindet:

Zu berechnen	Eingabe	Anzeige
101101:10	10 [÷]	**10110 b**
10110 AND 1111	1111 [f] [AND]	**110 b**

(Die Operation [÷] bewirkt das Einschalten der Statusanzeige **C**, die andeutet, daß der *Carryflag* gesetzt wurde. Flags werden auf Seite 36 erläutert. Sie können mit [g] [CF] 4 Flag und Statusanzeige wieder löschen.)

Alle vier Beispiele haben folgendes gemeinsam:

- Beide Zahlen werden vor der Operation eingegeben.
- [ENTER] wird nur zur Trennung zweier *hintereinander eingetasteter* Zahlen benötigt.
- Das Drücken einer Funktionstaste (hier [−], [×], [÷] und [AND]) bewirkt die sofortige Ausführung der Funktion und Anzeige des Ergebnisses.

Berechnungen im Gleitkomma-Modus

Im dezimalen Gleitkomma-Modus kann der HP-16C dezimale Gleitkomma-Arithmetik durchführen. Der Rechner wird durch die Funktion [FLOAT] in den Gleitkomma-Modus geschaltet und zeigt danach die spezifizierte Anzahl von Dezimalstellen an.

---

* Wenn Sie [f] [STATUS] drücken, sollte der Rechner **2-16-0000** anzeigen. Ist dies nicht der Fall, schlagen Sie auf Seite 37 nach.

Zu berechnen	Eingabe	Anzeige
(Gleitkomma dezimal)	[f] [FLOAT] 4 [BSP]	0.0000
–4.9:6	4.9 [CHS] [ENTER] 6 [÷]	–0.8167
√60	60 [g] [√x]	7.7460

Programmierung

Schreiben eines Programms. Der HP-16C wird programmiert, indem Sie einfach die Tastenfolge abspeichern, die Sie zur manuellen Lösung des Problems benutzen würden.

Beispiel: Schreiben Sie ein Iterationsprogramm, das zu einer gegebenen Zahl kontinuierlich Wert 1 hinzuaddiert.

Tastenfolge	Anzeige*	
[g] [P/R]	000–	Schaltet den Rechner in Programm-Modus (**PRGM** Statusanzeige erscheint). Zeile 000.
[f] CLEAR [PRGM]	000–	Löscht den Programmspeicher.
[g] [LBL] A	001-43,22, A	Programm erhält das Label «A».
1	002– 1	Zeile 002: 1.
[+]	003– 40	Zeile 003 addiert den Wert 1 zu dem beim Ablauf des Programms in der Anzeige befindlichen Wert.
[f] SHOW [BIN]	004– 42 26	Stoppt kurzzeitig und zeigt das Ergebnis in binärer Form.
[GTO] A	005– 22 A	Setzt Ausführung in einer Schleife fort.

* In der Anzeige erscheinen Zeilennummern und Tastencodes. Tastencodes sind zweistellige Zahlen, die die Reihen und Spaltenposition der gedrückten Taste bezeichnen.

Tastenfolge	Anzeige	
[g] [P/R]		Schaltet den Rechner zurück in den Run-Modus (die Statusanzeige **PRGM** erlischt). In der Anzeige erscheint das Ergebnis der letzten Berechnung.

Ausführung des Programms. Geben Sie einen Startwert (zum Beispiel Null) ein. Die Taste [ENTER] wird nicht benötigt, das Programm trennt die beiden Summanden automatisch und addiert den Wert 1 zu dem angegebenen Startwert.

Tastenfolge	Anzeige	
[DEC]		Schaltet auf Integer-Modus, Basis 10. (Sie können das Programm in jeder Zahlenbasis starten; das Ergebnis wird immer binär angezeigt.)
16 [f] [WSIZE]		Setzt Wortlänge auf 16.
0	**0 d**	Startwert 0.
[GSB] A	**1 b** **10 b** **11 b** **100 b** ⋮	Adressiert und startet das Programm mit Label «A». In der Anzeige erscheint das Ergebnis in binärer Form.
[R/S]	**22 d**	Da das Programm eine Endlosschleife enthält, müssen Sie die Programmausführung durch Drücken von [R/S] (*run/stop*) abbrechen. Die Anzeige enthält danach das Dezimaläquivalent der zum Zeitpunkt des Drückens von [R/S] gerade berechneten Binärzahl.

Diese Einführung in die Benutzung des HP-16C sollte Ihnen ein Gefühl für die Bedienung des Rechners vermitteln. In Teil I dieses Handbuchs, HP-16C Grundlagen, wird die Vielzahl der übrigen leistungsfähigen Funktionen des HP-16C vorgestellt.

Teil I
HP-16C
Grundlagen

Abschnitt 1

Der Einstieg

Dieser Abschnitt dient der detaillierten Einführung in die allgemeine Benutzung des HP-16C: Zahleneingabe, Löschen der Anzeige, Benutzung von [ENTER] und der Umgekehrten Polnischen Notation, Permanentspeicher. Den Beispielen liegt zwar der Integer-Modus zugrunde, aber die Arbeitsweise ist im dezimalen Gleitkomma-Modus identisch, falls anderes nicht explizit angemerkt ist. Dieser Abschnitt ist vor allem für Anwender gedacht, die nicht mit der Handhabung der derzeit erhältlichen HP-Rechner vertraut sind.

Ein- und Ausschalten

Die Taste [ON] schaltet den HP-16C ein und aus*. Um die Batterie zu schonen, schaltet sich der Rechner nach einigen Minuten selbsttätig aus, wenn keine Eingaben mehr erfolgen.

Tastenfeld-Operationen

Primär- und Alternativfunktionen

Die meisten Tasten Ihres HP-16C besitzen eine Primär- und zwei Alternativfunktionen**. Zur Ausführung der Primärfunktion brauchen Sie nur die entsprechende Taste zu drücken, zum Beispiel [÷]. Um die in goldfarbenen Buchstaben oberhalb, oder die in blauen Buchstaben unterhalb der Taste gekennzeichnete Alternativfunktion anzuwählen, ist zunächst die goldfarbene Vorwahltaste [f] bzw. die blaue Vorwahltaste [g] und anschließend die eigentliche Funktionstaste zu drücken; zum Beispiel [f][XOR] oder [g][DBL÷].

---

* Die Taste [ON] ist tiefer als die anderen Tasten angeordnet, um ein unbeabsichtigtes Einschalten zu verhindern.

** In diesem Handbuch werden wir uns bei der Bezugnahme auf Funktionen an bestimmte Konventionen halten. Die *Funktion selber* wird einfach durch ihren Namen in einem Rechteck repräsentiert. Die Benutzung einer *Funktionstaste* wird dagegen mit zugehöriger Vorwahltaste dargestellt, wie zum Beispiel: «Drücken Sie [g][MEM]». Bei Bezugnahme auf die «goldfarbenen» Funktionen unter den mit «CLEAR», «SET COMPL» oder «SHOW» bezeichneten Klammern werden den Funktionsnamen die Worte «CLEAR», «SET COMPL» oder «SHOW» vorausgestellt, wie zum Beispiel in «die Funktion CLEAR [REG]» oder «Drücken Sie [f] SHOW [DEC]».
Wenn einer Vorwahltaste verschiedene Tasten folgen können, werden die möglichen Tasten in geschweiften Klammern angegeben; zum Beispiel: «drücken Sie [f][WINDOW] ({0 bis 7}) ».

Beachten Sie, daß beim Drücken der Vorwahltasten [f] oder [g] die Statusanzeigen **f** oder **g** erscheinen und in der Anzeige stehenbleiben, bis Sie eine andere Taste drücken.

0 h
f

Löschen von Vorwahl-Eingaben

Eine Taste wird als Vorwahl-Taste bezeichnet, wenn zur Vollständigkeit der Tastenfolge einer Funktion noch weitere Tasten betätigt werden müssen. Anhang B enthält eine Liste aller Vorwahltasten.

Ist eine Vorwahltaste (wie z.B. [STO] oder [f] gedrückt worden, kann sie durch [f] CLEAR [PREFIX] wieder gelöscht werden; der Rechner ist danach zu neuen Eingaben bereit. Haben Sie versehentlich [f] anstelle von [g] oder umgekehrt gedrückt, korrigieren Sie dies einfach durch Drücken der anderen Taste.

Die «CLEAR»-Tasten

Mit Hilfe der verschiedenen «CLEAR»-Tasten können Sie die im folgenden aufgeführten Lösch-Operationen ausführen. Das Löschen eines Registers bedeutet hier, es mit dem Wert Null zu belegen.

Tastenfolge	Auswirkung
[f] CLEAR [PRGM]	
Im Run-Modus:	Der Rechner wird auf Zeile 000 des Programmspeichers gesetzt.
Im Programm-Modus:	Der gesamte Programmspeicher wird gelöscht.
[f] CLEAR [REG]	Löschen aller Datenspeicher-Register.
[f] CLEAR [PREFIX]	Löschen der Vorwahl einer unvollständig eingegebenen Tastenfolge.

Löschen der Anzeige: [CLx] und [BSP]

Der HP-16C besitzt zwei Funktionen zum Löschen der Anzeige: [CLx] (*clear X*) und [BSP] (*back space*).

Wirkung im Run-Modus:

- [CLx] löscht die Anzeige, die «0» erscheint.
- [BSP] entfernt nur die letzte angezeigte Ziffer, wenn die Zifferneingabe noch nicht abgeschlossen war. ([ENTER] und die meisten anderen Funktionen beenden die Zahleneingabe; die nächste eingegebene Ziffer ist Teil einer neuen Zahl.) Sie können dann die Eingabe mit der korrekten Ziffer fortsetzen. Bei beendeter Zifferneingabe wirkt [BSP] wie [CLx].

Tastenfolge	Anzeige	
[HEX]		Hex-Modus. Das Ergebnis der letzten Berechnung wird angezeigt.*
1234	1234 h	Zifferneingabe ist nicht abgeschlossen.
[BSP]	123 h	Löscht nur die letzte Ziffer.
1	1231 h	
[ENTER]	1231 h	Beendet die Zifferneingabe.
[BSP]	0 h	Löscht alle Ziffern.
12	12 h	
[g] [CLx]	0 h	Löscht die gesamte Anzeige, unabhängig vom Status der Zifferneingabe.

Wirkung im Programm-Modus:

- [CLx] ist programmierbar; daher wird die Funktion als Programmbefehl *abgespeichert* und löscht *nicht* den momentan angezeigten Befehl.
- [BSP] ist *nicht* programmierbar und kann daher zum Löschen von Programmbefehlen benutzt werden.

Funktionen einer Variablen

Funktionen einer einzigen Variablen operieren nur mit der Zahl in der Anzeige (X-Register). Zur Ausführung dieser Funktionen ist zunächst die als Argument zu verwendende Zahl in das X-Register einzugeben und anschließend die jeweilige Funktionstaste zu drücken.

Tastenfolge	Anzeige
0	0 h
[f] [NOT]	FFFF h

Funktionen zweier Variablen und [ENTER]

Eine Funktion zweier Variablen kann erst ausgeführt werden, wenn beide Argumente eingegeben sind. Beispiele für Funktionen zweier Variablen sind [+], [-], [×] und [÷].

* Wenn Sie [f][STATUS] drücken, sollte der Rechner **2-16-0000** anzeigen. Ist dies nicht der Fall, schlagen Sie auf Seite 37 nach.

Beenden der Zifferneingabe. Wenn eine Operation zwei Zahleneingaben erfordert, müssen Sie dem Rechner signalisieren, wann die Eingabe der ersten Zahl abgeschlossen ist. Dies geschieht mit Hilfe der Taste [ENTER]. Befindet sich jedoch eine der Zahlen schon im Rechner (als Ergebnis einer vorangegangenen Operation), wird der [ENTER]-Befehl nicht benötigt. Alle Tasten außer den Tasten zur Zifferneingabe selbst *beenden die Zifferneingabe.**

Kettenrechnungen. Lange Berechnungen können ohne Verwendung von Klammern ausgeführt werden, in dem Sie zwei nacheinander in den Stack zu ladende Zahlen durch [ENTER] trennen.

Beispiel: Berechnen Sie (6 + 7) × (9 – 3) im Dezimalsystem.

Tastenfolge	**Anzeige**	
[DEC]		Dezimal-Modus. In der Anzeige erscheint das zuletzt berechnete Ergebnis.
6 [ENTER]	6 d	Zahleneingabe beendet.
7 [+]	13 d	«13» wird als Zwischenergebnis abgespeichert.
9 [ENTER]	9 d	
3 [–]	6 d	«6» wird ebenfalls als Zwischenergebnis gespeichert.
[×]	78 d	(13 × 6) = 78

Der Permanentspeicher

Vom Permanentspeicher erhaltene Informationen

Der Permanentspeicher des HP-16C sorgt auch nach dem Ausschalten für die Erhaltung folgender Informationen im Rechner:

- Zahlenbasis oder Arbeitsmodus (Hexadezimal, Dezimal, Oktal, Dual oder dezimales Gleitkomma).
- Arithmetischer Modus (Einerkomplement, Zweierkomplement, Unsigned)
- Wortlänge.

* Die Tasten zur Zifferneingabe bestehen aus den Zifferntasten und der Taste [BSP]. Im dezimalen Gleitkomma-Modus kommen zusätzlich die Tasten [.], [EEX] und [CHS] hinzu.

- Alle gespeicherten Zahlen.
- Alle abgespeicherten Programme.
- Momentane Position des Rechners im Programmspeicher.
- Alle nicht abgearbeiteten Unterprogrammrücksprünge.
- Flag-Zustände.
- Position des Anzeige-Fensters.
- Momentanes Dezimaltrennzeichen.

Beim Einschalten «erwacht» der Rechner immer im Run-Modus. Zur Eingabe eines Programms müssen Sie den Rechner zuvor explizit durch [g][P/R] in den Programm-Modus schalten.

Bei ausgeschaltetem Rechner bleibt der Permanentspeicher auch ohne Batterien für kurze Zeit erhalten. Das Auswechseln der Batterien wird in Anhang C beschrieben.

Löschen des Permanentspeichers

Der Permanentspeicher kann wie folgt gelöscht werden:*

1. Schalten Sie den Rechner aus.
2. Halten Sie die Taste [ON] gedrückt, und drücken Sie zusätzlich die Taste [-].
3. Lassen Sie zuerst die Taste [ON], dann [-] wieder los. (Schritt 2 und 3 werden in diesem Handbuch durch [ON]/[-] symbolisiert.)

Fehleranzeige. Nach einer Löschung des Permanentspeichers wird die Meldung **Pr Error** (*power error*) angezeigt. Diese Anzeige kann durch Drücken einer beliebigen Taste gelöscht werden. Anhang A enthält eine Auflistung der Fehlermeldungen und der Fehlerbedingungen. Ursachen.

Voreinstellungen. Beim erstmaligen Einschalten oder nach dem Löschen des Permanentspeichers sind folgende Voreinstellungen wirksam:

- Zahlenbasis: Hexadezimal (Integer-Modus).
- Zweierkomplement-Modus.
- 16 Bit Wortlänge.
- Alle Flags gelöscht.
- Programmspeicher und alle Register gelöscht.

---

* Wird der Rechner starken Erschütterungen ausgesetzt, kann dies ein Löschen des Permanentspeichers zur Folge haben.

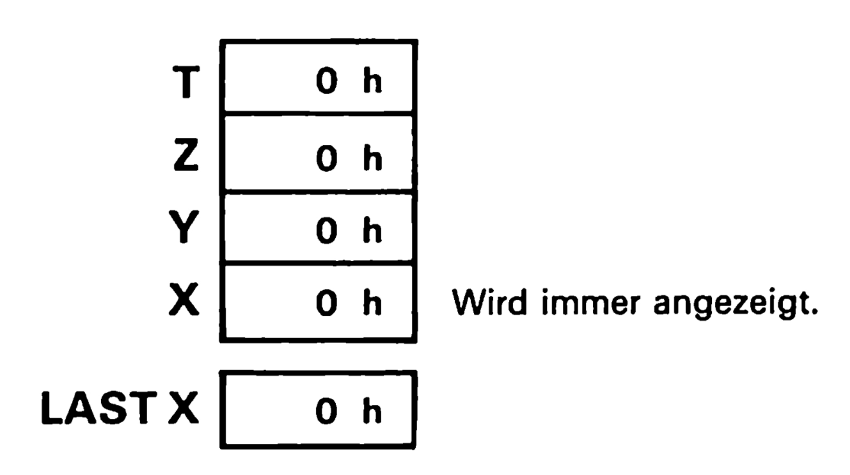
Abschnitt 2

Der automatische Speicherstack

Speicherstack und Stackmanipulation

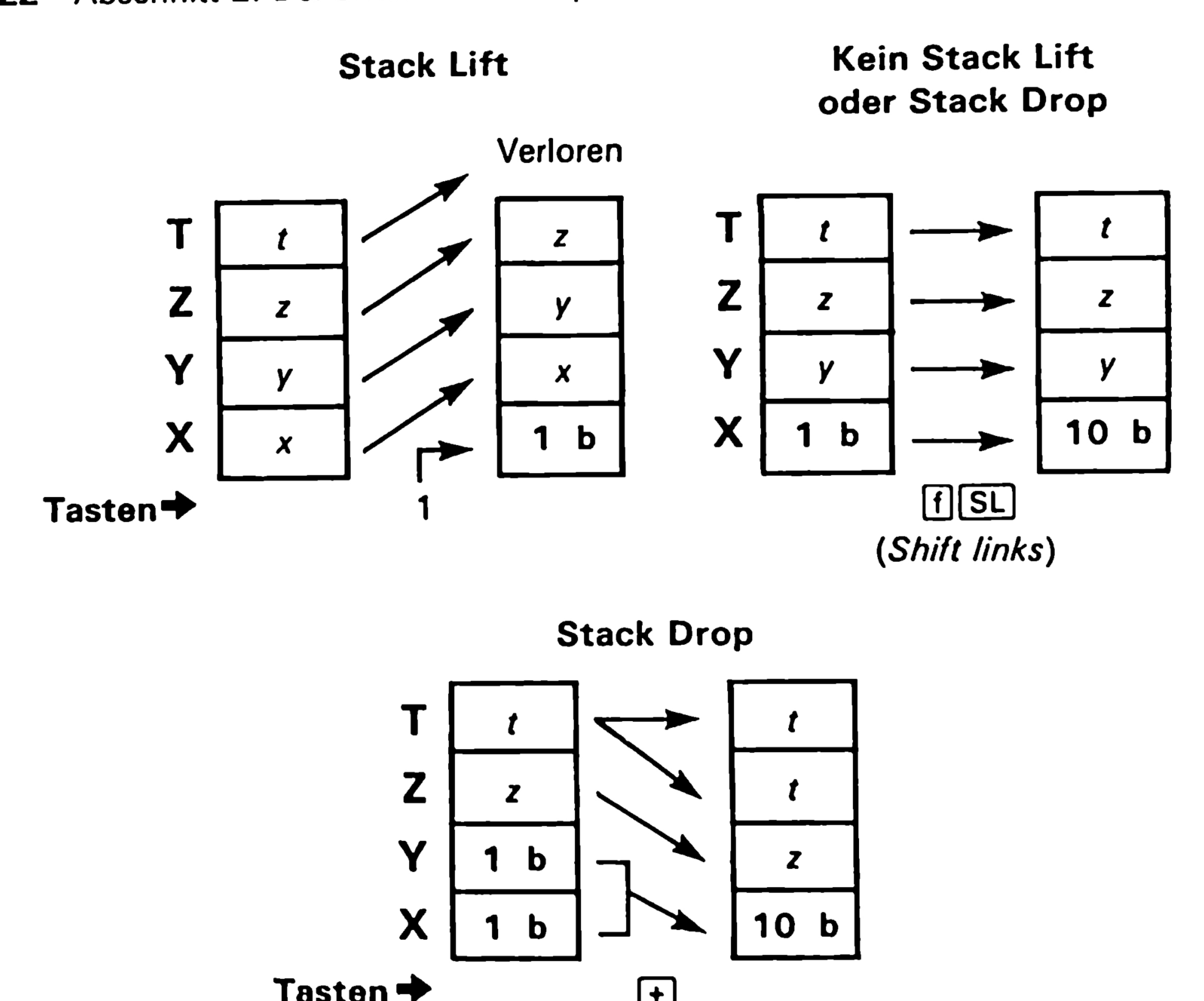
Der HP-16C arbeitet mit Umgekehrter Polnischer Notation, um komplizierte Berechnungen ohne Klammern und mit möglichst wenigen Tasten durchführen zu können. Durch den Speicherstack und die [ENTER]-Taste werden Zwischenergebnisse automatisch gespeichert und zurückgerufen. In diesem Abschnitt wird die Arbeitsweise des Rechner-Stacks erläutert, die für die Benutzung des HP-16C von grundlegender Bedeutung ist.

Register des automatischen Speicherstacks



Solange sich der Rechner nicht im Programm-Modus befindet, steht in der Anzeige immer der Wert des X-Registers.

Die Zahlen im Stack sind auf der Basis last in/first out verfügbar. Die folgende Illustration veranschaulicht die drei möglichen Stack-Bewegungen. Der Rechner sei dabei im Binär-Modus, und x, y, t und z seien beliebige Zahlen im Stack.

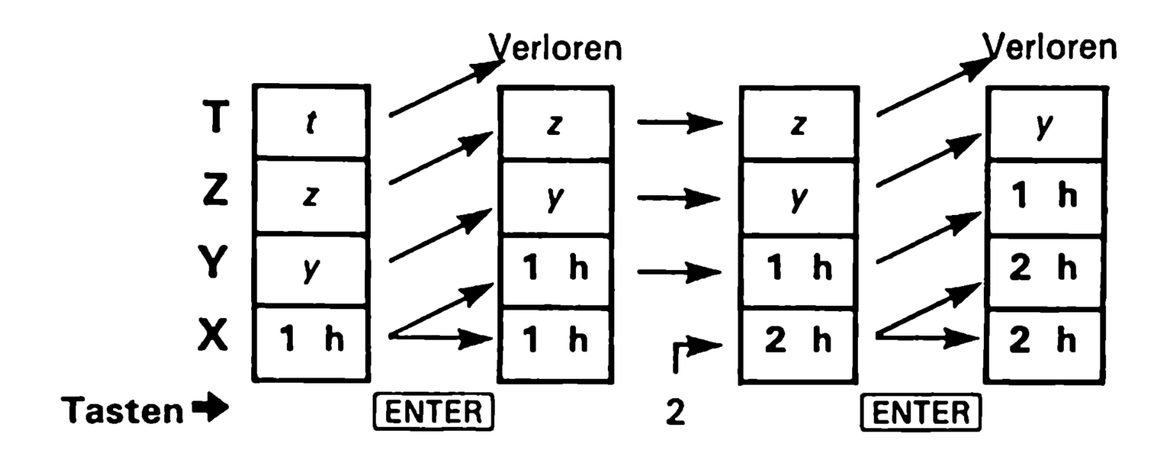


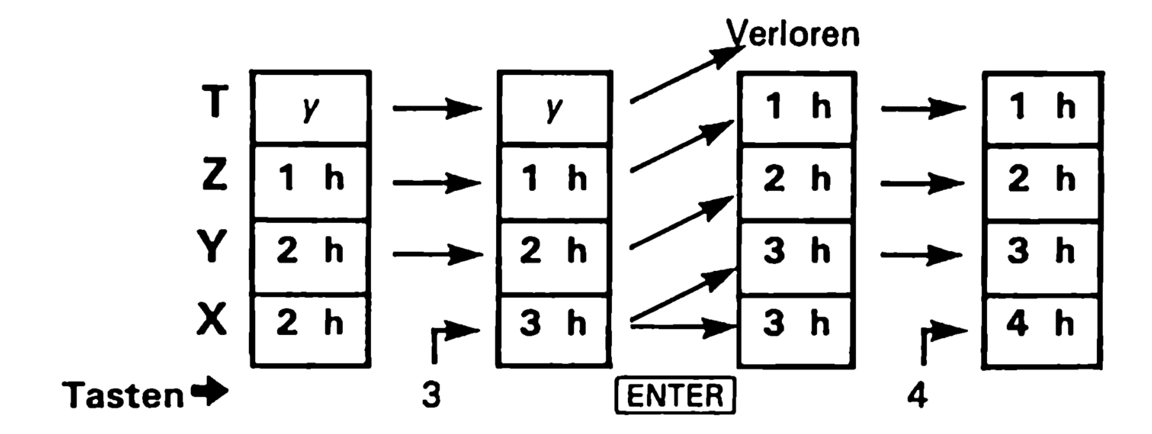
Normalerweise bedingen Funktionen einer Variablen keine Bewegung des Stacks, während Funktionen zweier Variablen *in der Regel* ein Herunterfallen des Stacks (einen Stack Drop) auslösen.

Beim Stack Drop wird die Zahl im T-Register reproduziert; damit kann dieser Wert als automatische Konstante benutzt werden.

Funktionen zur Stackmanipulation

Wie bereits erwähnt, trennt [ENTER] zwei nacheinander eingegebene Zahlen. Wenn [ENTER] gedrückt wird, verschiebt der Rechner den Stack-Inhalt nach oben (*Stack Lift*), indem er die Zahl im angezeigten X-Register ins Y-Register kopiert. Die drauffolgende Eingabe überschreibt den Wert im X-Register; es erfolgt kein Stack Lift. Das folgende Beispiel illustriert die Stackverschiebung bei der Belegung des Stacks mit den Zahlen 1, 2, 3 und 4. (Schattierte Felder deuten an, daß der Inhalt dieses Registers beim Eingeben oder Abrufen der nächsten Zahl überschrieben wird.)





Zusätzlich zu [ENTER] verschieben drei weitere Funktionen die Stackinhalte:

- [R↓] (*roll down*) verschiebt die Inhalte des Stack zyklisch um ein Register nach unten; die Zahl im X-Register wird ins T-Register übertragen.
- [R↑] (*roll up*) verschiebt die Stackinhalte zyklisch um ein Register nach oben; die Zahl im T-Register wird ins X-Register übertragen.
- [x≷y] (*X exchange Y*) vertauscht die Inhalte des X- und Y-Registers.

Das Register LAST X

Bei der Ausführung einer numerischen Funktion wird der Wert, der das angezeigte X-Register vor der Funktionsausführung belegt hat, in das Register LAST X gerettet*. Durch Drücken von [g][LSTx] wird der aktuelle Inhalt des LAST X-Registers wieder in das angezeigte X-Register zurück übertragen.

* Anhang B enthält eine komplette Liste der Operationen, die *x* in das LAST X-Register retten.

Mit Hilfe von [LSTx] können Sie einen konstanten Wert wiederholt benutzen, ohne ihn erneut eintasten zu müssen (siehe «Berechnungen mit Konstanten», Seite 26). Sie können [LSTx] auch zur Fehlerkorrektur benutzen, indem Sie damit die Zahl ins X-Register zurückrufen, die dort vor der zuletzt ausgeführten numerischen Funktion abgespeichert war.

Nehmen Sie zum Beispiel an, daß Sie in einer Kettenrechnung irrtümlicherweise einen falschen Summanden (10 anstatt 11) eingegeben haben:

Tastenfolge	Anzeige	
[BIN]		Binär-Modus. Anzeige enthält das Ergebnis der letzten Berechnung.
1010 [ENTER]	**1010 b**	
10 [+]	**1100 b**	Falsche Eingabe!
[g] [LSTx]	**10 b**	Ruft aus dem LAST X-Register die letzte Eingabe ins X-Register vor der Ausführung von [+] ab.
[−]	**1010 b**	Annulliert die Wirkung der Funktion, die das falsche Ergebnis geliefert hat.
11 [+]	**1101 b**	Richtiges Ergebnis.

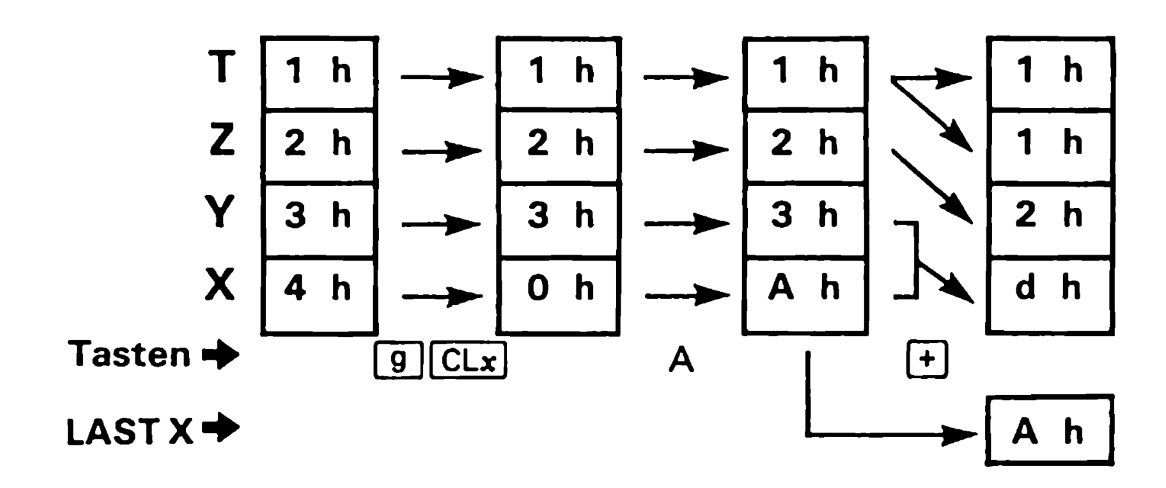
Numerische Funktionen und Stack

Stack-Verschiebungen

Wenn Sie zwei Zahlen nacheinander eingeben wollen, müssen Sie zur Trennung der beiden Zahlen [ENTER] drücken. Wenn Sie jedoch nur eine Zahl eingeben wollen, weil sich die andere Zahl bereits im angezeigten X-Register als Ergebnis einer vorangegangenen Berechnung oder sonstigen Funktionsausführung (wie etwa [R↓] befindet, brauchen Sie [ENTER] nicht zu bestätigen, Die Ausführung der meisten HP-16C Funktionen hat den folgenden Zusatzeffekt:

- Der automatische Speicherstack wird freigegeben, d.h. bei Eingabe oder Abruf der *nächsten* Zahl verschiebt sich der Stackinhalt automatisch nach oben.
- Die Zifferneingabe wird beendet; die nächste eingegebene Ziffer ist Teil einer neuen Zahl.

Zwei Funktionen – [ENTER] und [CLx] – *sperren* den Stack, d.h. der Stack wird bei der Eingabe der *nächsten* Zahl nicht nach oben verschoben. Nach Ausführung einer der genannten Funktionen wird die momentan angezeigte Zahl bei der Eingabe eines neuen Wertes überschrieben. (Es erfolgt zwar ein Stack Lift, wenn Sie [ENTER] drücken, jedoch *nicht* bei der Eingabe oder dem Abruf der *nächsten* Zahl.)



Wie Sie aus der Illustration ersehen können, wird bei der Ausführung einer arithmetischen Operation mit den Operanden (A_{16} und 3_{16}) im X- bzw. Y-Register der Stack nach unten verschoben und das Ergebnis (D_{16}) im X-Register abgelegt.

Anhang B enthält eine komplette Auflistung, wie der Stack Lift und die Zifferneingabe von den Funktionen des HP-16C beeinflusst werden.

Kettenrechnungen

Der automatische Stack Lift und Drop ermöglicht Kettenrechnungen ohne Verwendung von Klammern. Zwischenergebnisse werden automatisch im Stack gespeichert und nach Bedarf zurückgerufen. Kettenrechnungen werden einfach als Serie von Funktionen einer und zweier Variablen abgearbeitet. Wenn Sie mit Ihrer Berechnung bei der innersten Zahl oder Klammer beginnen und nach außen arbeiten (wie Sie es auch mit Papier und Bleistift tun würden), werden Sie selten Zwischenergebnisse in einem Datenregister speichern müssen.

* Analog zu [CLx] sperrt auch [BSP] den Stack Lift und löscht die Anzeige, sofern die Ziffemeingabe abgeschlossen wurde. Andernfalls wird der Stack Lift weder freigegeben noch gesperrt.

Betrachten Sie die folgende Berechnung im dezimalen Integer-Modus:

$$3[4+5(6+7)]$$

Tastenfolge	Anzeige	
[DEC]		Anzeige enthält das Ergebnis der letzten Berechnung.
6 [ENTER] 7 [+]	13 d	Zwischenergebnis.
5 [×]	65 d	Zwischenergebnis.
4 [+]	69 d	Zwischenergebnis.
3 [×]	207 d	Endergebnis.

Dieses Beispiel zeigt, daß nach jeder Operation mit zwei Zahlen der Stack automatisch abgesenkt und bei jeder nachfolgenden Eingabe einer Zahl wieder angehoben wird.

Berechnungen mit Konstanten

Sie können auf zwei Arten wiederholte Berechnungen mit einer Konstanten durchführen, ohne dabei ein Datenspeicher-Register verwenden zu müssen:

- Sie verwenden das LAST X-Register.
- Sie laden den Stack mit der Konstanten, bevor Sie die Berechnungen durchführen.

Beispiel: Entfernen Sie die höherwertigen vier Bits unter Erhaltung der niederwertigen vier Bits der folgenden 8-Bit Binärzahlen: 10001001, 10101111 und 11110101. Die Konstante 1111 dient dabei als Maske.

Verwendung des LAST X-Registers. Führen Sie die Berechnung mit der Konstanten im X-Register (und nicht im Y-Register) durch, so daß sie immer in das LAST X-Register gerettet wird. Mit [g][LSTx] können Sie die Konstante dann immer wieder zurückladen.

Tastenfolge	Anzeige	
[BIN]		Binär-Modus. Anzeige enthält das Ergebnis der letzten Berechnung.
10001001 [ENTER]	10001001 b	Erste Zahl.
1111	1111 b	Maske (die Konstante).
[f] [AND]	1001 b	Niederwertige vier Bits.

Tastenfolge	Anzeige	
10101111	**10101111 b**	Zweite Zahl.
[g] [LSTx]	**1111 b**	Zurückladen der Konstanten.
[f] [AND]	**1111 b**	Niederwertige vier Bits.
11110101 [g] [LSTx]	**1111 b**	
[f] [AND]	**101 b**	Niederwertige vier Bits.

Verwendung des Stack. Laden Sie den Stack mit einer Konstanten, indem Sie diese eingeben und dann dreimal nacheinander [ENTER] drücken. Nach jeder Operation (hier: [AND]) erfolgt ein Stack Drop (die Konstante ist dann im Y-Register verfügbar), bei dem die Konstante im T-Register reproduziert wird. Wenn Sie mit [BSP] oder [CLx] den Stack sperren, wird mit der neu eingegebenen Variablen das vorangegangene Ergebnis überschrieben und die Konstante bleibt erhalten.

Tastenfolge	Anzeige	
1111 [ENTER]	**1111 b**	Maske (die Konstante).
[ENTER] [ENTER]	**1111 b**	Füllt den Stack mit 1111.
10001001 [f] [AND]	**1001 b**	Niederwertige vier Bits der ersten Zahl.
[BSP]	**0 b**	Stack Lift gesperrt.
10101111 [f] [AND]	**1111 b**	Niederwertige vier Bits der zweiten Zahl.
[BSP]	**0 b**	Stack Lift gesperrt.
11110101 [f] [AND]	**101 b**	Niederwertige vier Bits der dritten Zahl.

Abschnitt 3

Zahlensysteme und Anzeige-Formatierung

Die Zahlendarstellung im HP-16C ist sehr viel vielseitiger als bei anderen Rechnern. Dieser Abschnitt behandelt die verschiedenen Aspekte der Verwendung und Anzeige von ganzzahligen Werten: Zahlensysteme, Wortlänge, Komplemente, Zahlenbereiche, resultierende Anzeigen. Dieser Abschnitt beschreibt nur die Formate im Integer-Modus; das Gleitkomma- Format wird in Abschnitt 5 behandelt. Der Permanentspeicher erhält das jeweils gewählte Format.

Integer-Modus

Die Zahlensysteme ([HEX], [DEC], [OCT] und [BIN]) arbeiten ausschließlich im Integer-Modus. Gebrochene Dezimalzahlen sind nur im dezimalen Gleitkomma-Modus zulässig (siehe Abschnitt 5). Durch Drücken einer der vier Zahlenbasis-Tasten wird der Rechner automatisch in den Integer-Modus geschaltet.

Zahlenbasen

Der HP-16C besitzt vier Zahlenbasis-Einstellungen im Integer-Modus: Hexadezimal (Basis 16), Dezimal (Basis 10), Oktal (Basis 8) und Dual oder Binär (Basis 2). Ein **h**, **d**, **o** oder **b** rechts in der achtstelligen Anzeige deutet die jeweilige Zahlenbasis an. Beim ersten Einschalten oder nach einem Löschen des Permanentspeichers ist der Rechner auf Hexadezimal voreingestellt.

Durch Drücken von [HEX], [DEC], [OCT] oder [BIN] wird die angezeigte Zahl in den äquivalenten Wert im neu gewählten Zahlensystem umgewandelt und in einem rechtsbündigen Integer-Format angezeigt.Danach gedrückte Zifferntasten werden dem Modus entsprechend interpretiert; der Rechner reagiert nicht, wenn Sie versuchen, eine unzulässige Ziffer (wie z.B. eine «3» im Dual-Modus) einzugeben. Im Hexadezimal-Modus werden zusätzlich zu den Zifferntasten [0] bis [9] auch die Tasten [A] bis [F] zur Zifferndarstellung verwendet und in der Form **A**, **b**, **C**, **d**, **E** und **F** angezeigt.

Hinweis: Unabhängig von der momentan eingestellten Zahlenbasis *ist die interne Zahlendarstellung immer binär.* Durch Umschalten der Zahlenbasis ändert sich nur die Anzeige, nicht die rechnerinterne Darstellung des Wertes*.

Kurzzeitige Anzeige («SHOW»)

Um den Anzeigewert kurzzeitig in einer anderen Basis zu betrachten, drücken Sie [f] SHOW {[HEX], [DEC], [OCT], [BIN]}. Die umgewandelte Form der Zahl wird nur solange angezeigt, wie Sie die Zahlenbasis-Taste gedrückt halten.

Tastenfolge	Anzeige
[HEX] F	F h
[BIN]	1111 b
[f] SHOW [OCT] (halten)	17 o
(loslassen)	1111 b

Komplement-Modi und Unsigned-Modus

Der HP-16C verfügt über drei Konventionen zur Zahlendarstellung: *Einerkomplement*, *Zweierkomplement* und *Unsigned* (vorzeichenlos).

Beim ersten Einschalten und nach einem Löschen des Permanentspeichers ist der Rechner auf Zweierkomplement-Modus voreingestellt. Ein gesetzter Modus bleibt so lange erhalten, bis Sie ihn explizit ändern oder den Permanentspeicher löschen. (Alle Beispiele in diesem Handbuch verwenden den Zweierkomplement-Modus, falls nicht anders vermerkt.)

In der Binärdarstellung einer Zahl mit Vorzeichen wird das linke oder höchstwertige Bit (entsprechend der momentanen Wortlänge) als Vorzeichen bit benutzt: 0 wird als plus und 1 als minus interpretiert. Im Dezimal-Modus wird eine negative Zahl mit einem Minuszeichen angezeigt.

---

* Die Tastenfolge auf Seite 35 zeigt, wie Zahlenbasis, Wortlänge und Komplementmodus auf die Anzeige wirken, ohne die interne Binärdarstellung einer Zahl im Integer-Modus zu beeinflussen.

Einerkomplement-Modus

Die Tastenfolge [f] SET COMPL [1's] schaltet den Rechner in den Einerkomplement-Modus. In diesem Modus bewirkt das Drücken der Taste [CHS] (*change sign*) die Komplementierung aller Bits der Zahl in der Anzeige (Einerkomplement der Zahl).

Im Einerkomplementmodus sind gleichviele positive wie negative Zahlen darstellbar; die Null hat zwei Darstellungen: +0 und −0.

Zweierkomplement-Modus

Die Tastenfolge [f] SET COMPL [2's] schaltet in den Zweierkomplement-Modus. In diesem Modus bildet die Funktion [CHS] das Zweierkomplement der Zahl in der Anzeige (alle Bits des X-Registers werden komplementiert, anschließend wird 1 addiert).

Im Zweierkomplementmodus gibt es nur eine Darstellung für die Null, aber der darstellbare positive Zahlenbereich enthält immer eine Zahl weniger als der darstellbare negative Zahlenbereich.

Unsigned-Modus

Die Tastenfolge [f] SET COMPL [UNSGN] setzt den Unsigned-Modus, in dem kein Vorzeichen benutzt wird. In diesem Modus hat die erste Ziffer einer Zahl Stellenwert und nicht die Bedeutung eines Vorzeichens. Die größte durch ein 8-Bit Wort darstellbare Zahl ist also 255_{10} und nicht 127_{10}.

Ein Vorzeichenwechsel hat im Unsigned-Modus keine Auswirkung. Wenn Sie [CHS] in diesem Modus drücken, erhalten Sie das Zweierkomplement der Zahl in der Anzeige. Flag 5 (durch die Statusanzeige **G** repräsentiert) wird gesetzt, um anzudeuten, daß das Ergebnis in Wirklichkeit negativ ist und damit außerhalb des Zahlenbereichs im Unsigned-Modus liegt.

Die folgende Tabelle enthält die jeweiligen Dezimalwerte aller möglichen 4-Bit Kombinationen (bei Wortlänge 4) in den drei Komplement-Modi.

Dezimalwerte aller 4-Bit Binärkombinationen

Binär	Einerkomplement-Modus	Zweier-komplement-Modus	Unsigned-Modus
0111	7	7	7
0110	6	6	6
0101	5	5	5
0100	4	4	4
0011	3	3	3
0010	2	2	2
0001	1	1	1
0000	0	0	0
1111	-0	-1	15
1110	-1	-2	14
1101	-2	-3	13
1100	-3	-4	12
1011	-4	-5	11
1010	-5	-6	10
1001	-6	-7	9
1000	-7	-8	8

Wortlänge und Anzeigefenster

Der HP-16C kann mit bis zu 64 Bit langen *Worten* arbeiten. Die Wortlänge ist beim ersten Einschalten oder einem Löschen des Permanentspeichers auf 16 Bit voreingestellt. Das Anzeigefenster zeigt jeweils acht Bit gleichzeitig an, führende Nullen werden unterdrückt*. Ein Punkt rechts oder links der Statusanzeigen **h**, **d**, **o** oder **b** signalisiert das Vorhandensein weiterer, nicht angezeigter Stellen rechts oder links des derzeitig angezeigten Teils der Zahl.

* Bei gesetztem Flag 3 (siehe Seite 36) werden alle führenden Nullen mit angezeigt.

Wortlänge

Zur Spezifikation einer Wortlänge geben Sie die gewünschte Wortlänge (1_{10} bis 64_{10}) in das X-Register ein und drücken anschließend [f] [WSIZE]. Dabei ist der Absolutwert der Zahl im X-Register maßgebend; eine Null wird als Wortlänge 64 interpretiert. Die Ausführung von [WSIZE] beinhaltet einen Stack Drop.

Bei einer momentanen Wortlänge von weniger als 8 Bit ist die zulässige Größe der Zahl zur Definition einer neuen Wortlänge begrenzt; Sie können diese Einschränkung jedoch jederzeit umgehen, indem Sie die Wortlänge mit [f] [WSIZE] auf 64 Bit setzen. (Anschließend können Sie jede beliebige Wortlänge wählen.) Auf den Versuch, eine Wortlänge größer als 64 einzugeben, erfolgt die Fehlermeldung **Error 2***.

Tastenfolge	Anzeige	
[DEC] 16 [f] [WSIZE]		Basis 10, Wortlänge 16.
[f] SET COMPL [2's]		Zweierkomplement-Modus gesetzt.
32767 [ENTER]	**32767 d**	Größte positive Zahl im Zweierkomplementmodus mit Wortlänge 16.
8 [f] [WSIZE]	**-1 d**	Zahl wird von 01111111 11111111_2 (16 Bit) in 11111111_2 (acht Bit) umgewandelt.
16 [f] [WSIZE]	**255 d**	Die Zahl 11111111_2 wird zu 00000000 11111111_2.

Hinweis: Das Ändern der Wortlänge kann numerisch äquivalente Werte *im Stack* verfälschen. Die Wahl einer kleineren Wortlänge bewirkt ein Abschneiden des Wortes, nur die niederwertigen Bits bleiben erhalten. Beim Vergrößern der Wortlänge wird das Vorzeichenbit einer negativen Zahl nicht berücksichtigt. Diese Änderungen werden durch Wiederherstellung der ursprünglichen Wortlänge nicht wieder rückgängig gemacht. Im Gegensatz zu den Stackregistern hat eine Änderung der Wortlänge keine Auswirkung auf die Datenspeicher-Register (siehe Seite 67).

---

* Im Einer- und Zweierkomplementmodus erhalten Sie möglicherweise eine negative Zahl, wenn Sie versuchen, eine Zahl einzugeben, die mit der momentanen Wortlänge nicht dargestellt werden kann. Dies ist dann der Fall, wenn das höchstwertige Bit (das Vorzeichenbit) den Wert 1 (negativ) annimmt, wie am Ende der Tastenfolge auf Seite 36 gezeigt wird.

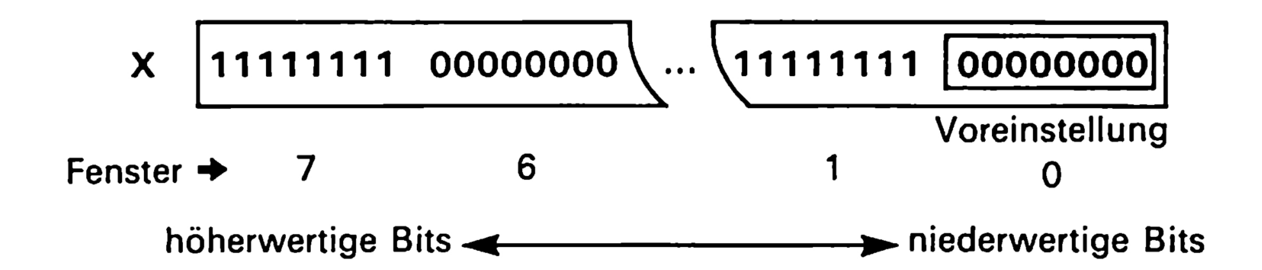
Bei Wortlängen von 3 oder kleiner bewirkt der Versuch, eine für die *gegebene Wortlänge zu große Ziffer* (auch wenn sie in der derzeitigen Zahlenbasis erlaubt ist) einzugeben, ein Ersetzen der ursprünglichen Eingabe durch eine Null.

Fenster

Die Anzeige kann als *Fenster* betrachtet werden, das bis zu acht Stellen der Zahl im X-Register sichtbar macht. Das X-Register kann wie alle Register, in Abhängigkeit von der Wortlänge, bis zu 64 Binärstellen speichern. Normalerweise betrachten Sie Fenster 0 mit den acht niederwertigen Stellen der Zahl im X-Register. Bei der Eingabe von mehr als acht Ziffern wandern die höherwertigen Stellen links aus der Anzeige und in Fenster 1.

Durch die Tastenfolge [f] [WINDOW] {0 bis 7} können Sie verschiedene Segmente der Zahl im X-Register zur Anzeige bringen. Nach jeder Neueingabe ins X-Register wird wieder Fenster 0 angezeigt. Die größte Fenster-Nummer ist 7, da die maximale Wortlänge 64 beträgt. (Bei kleineren Wortlängen oder Zahlen sind die höhernumerierten Fenster leer). Das Anwählen eines Fensters größer als 7 bewirkt die Anzeige der Fehlermeldung **Error 1**.

Beispiel: Die 16-stellige Hexadezimalzahl FF00 FF00 FF00 FF00 besitzt eine 64-stellige Binärdarstellung (abwechselnd 8 Einsen, 8 Nullen). Im Binär-Modus können Sie die ganze Zahl anzeigen, indem Sie nacheinander [f] [WINDOW] 0 bis [f] [WINDOW] 7 ausführen.

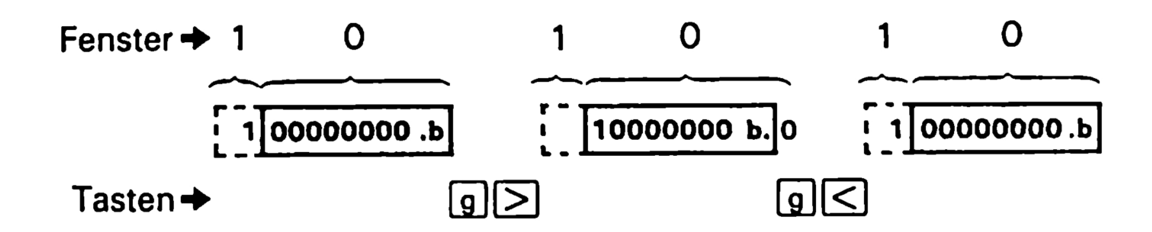


Verschieben des Fensters

Die Tasten [<] und [>] verschieben das Fenster und bringen somit verschiedene Teile der Zahl im X-Register zur Anzeige. Durch Drücken von [<] und [>] wird das Fenster um jeweils eine Stelle nach rechts bzw. nach links verschoben. Dies verändert nicht die Zahl im X-Register, sondern bringt nur einen anderen Teil davon zur Anzeige.

Die Position des Punktes zeigt an, wo sich der Rest der Zahl im X-Register befindet. Steht der Punkt z.B. links des Basis-Indikators (**.b**), dann sind noch weitere Stellen links der momentanen Anzeige vorhanden. Die Tastenfolge [g] [>] bewirkt Verschieben des Fensters nach links, diese «verborgenen» Stellen erscheinen dadurch in der Anzeige.

Ein Punkt kann auf beiden Seiten des Basis-Indikators auftreten, falls das derzeitige Fenster nicht das rechte oder linke Ende der Zahl anzeigt.



Beispiel: Mit den folgenden Verschiebe- und [WINDOW]-Funktionen können Sie den gesamten Inhalt des X-Registers zur Anzeige bringen. In diesem Beispiel wird eine Wortlänge von 16 Bit unterstellt.

Tastenfolge	**Anzeige**	
[BIN]		Setzt Binär-Modus.
11111111 [ENTER]	11111111 b	Anzeige gefüllt (8 Stellen).
1 [+]	00000000 .b	Punkt steht links, d.h. weitere Stellen links.
[g] [>]	10000000 b.	Schiebt die Zahl um eine Stelle nach rechts; weitere Stellen rechts.
[f] [WINDOW] 1	1 b.	Inhalt von Fenster 1; die höchstwertige Stelle.
[f] [WINDOW] 0	00000000 .b	Inhalt von Fenster 0; die niedrigstwertige Stelle.

Eine Verschiebung des Fensters wird nach Ausführung einer Bit-Operation oder einer mathematischen Funktion wieder aufgehoben, d.h. die Anzeige wird auf Fenster 0 zurückgesetzt. Eine vollständige Auflistung der Funktionen, die die Anzeige *nicht* auf Fenster 0 zurücksetzen, finden Sie in Anhang B.

Anzeige und interne Darstellung

Die folgenden Tastenfolgen veranschaulichen, wie die verschiedenen Modi (Zahlenbasis, Wortlänge und Komplement) auf die Anzeige und die interne Darstellung wirken.

Tastenfolge	Anzeige	Interne Binärdarstellung
[HEX]		
8 [f] [WSIZE]		
[BSP]	**0 h**	00000000
62	**62 h**	01100010
[OCT]	**142 o**	01100010
[BIN]	**1100010 b**	01100010
[DEC]	**98 d**	01100010
62	**62 d**	00111110
[OCT]	**76 o**	00111110
62	**62 o**	00110010
[HEX]	**32 h**	00110010
[f] SET COMPL [2's]	**32 h**	00110010
[CHS]	**CE h**	11001110 Negative Zahl im Zweierkomplement-Modus bei Wortlänge 8.
[OCT]	**316 o**	11001110
[BIN]	**11001110 b**	11001110
[DEC]	**−50 d**	11001110
[f] SET COMPL [1's]	**−49 d**	11001110 Interne Darstellung unverändert.
[f] SET COMPL [UNSGN]	**206 d**	11001110
[f] SET COMPL [1's]	**−49 d**	11001110 Wird im Einerkomplement-Modus als negative Zahl interpretiert.
[CHS]	**49 d**	00110001

Tastenfolge	**Anzeige**	**Interne Binärdarstellung**
2	**2 d**	00000010
5	**25 d**	00011001
4	**–001 d**	11111110 (entspricht -1_{10} im Einerkomplement; die Nullen sind Platzhalter für Korrekturen bei Zifferneingabe).
[f] SET COMPL [UNSGN]	**254 d**	11111110

Flags

Der HP-16C verfügt über drei *Benutzer*-Flags (0, 1 und 2), die zur Kontrolle der Programmausführung benutzt werden können, und drei *System*-Flags, die bestimmte Systemzustände kennzeichnen bzw. steuern.

Der Gebrauch von Flags bei der Programmierung wird in Abschnitt 9, «Programmverzweigungen und Programmsteuerung» behandelt. Die Flags 3, 4 und 5 sind einfach Statusanzeigen und haben keine Wirkung auf die Rechnerfunktion (falls Sie sie nicht zur Kontrolle der Programmausführung benutzen wollen):

- Flag 3 steuert die Anzeige führender Nullen. Ist er gesetzt, werden auch Nullen links der höchstwertigen von Null verschiedenen Ziffer angezeigt. Ist er gelöscht (Voreinstellung), werden führende Nullen in der Anzeige unterdrückt. (Beachten Sie, daß im Dezimal- und im dezimalen Gleitkomma-Modus führende Nullen immer unterdrückt werden.

- Flag 4 wird gesetzt (und die Statusanzeige **C** erscheint), wenn eine Carry Bedingung (*Übertrag*-Bedingung) auftritt (siehe Seite 39).

- Flag 5 wird gesetzt (und die Statusanzeige **G** erscheint), wenn eine **Out-Of-Range** Bedingung auftritt, d.h. wenn der von einer Operation zurückgegebene Wert größer als die größte darstellbare Zahl oder im momentanen Modus nicht darstellbar ist.

In Abschnitt 4 wird erläutert, wann Carry und Out-Of-Range Bedingungen auftreten.

Alle Flags können wie folgt gesetzt, gelöscht und abgefragt werden:

- [g] [SF] *n* setzt Flag *n* (0 bis 5).
- [g] [CF] *n* löscht Flag *n* (0 bis 5).
- [g] [F?] *n* fragt den Zustand von Flag *n* ab.

Der Zustand eines Flags und gegebenenfalls zugehörigen Statusanzeige bleibt solange erhalten, bis er durch eine der folgenden Operationen geändert wird:

- Löschen des Permanentspeichers.
- Ausführen einer Funktion, die den entsprechenden Flag verändert (nur Flag 4 und 5).
- Setzen oder Löschen des Flags mit [SF] oder [CF].

Bei der Programmierung werden Flags gewöhnlich benutzt, um das Ergebnis einer Abfrage für späteren Gebrauch abzuspeichern. Abschnitt 9 beschreibt die Verwendung von Flags bei bedingten Verzweigungen.

Rechner-Status ([STATUS])

Das Drücken von [f] [STATUS] bewirkt die vorübergehende Anzeige 1) des derzeitigen Komplement-Modus, 2) der momentanen Wortlänge und 3) der derzeit gesetzten Flags (nur Flags 0 bis 3; die Flags 4 und 5 werden durch die Statusanzeigen **C** und **G** repräsentiert). Der Rechner-Status wird nur solange angezeigt, wie Sie die Taste [STATUS] gedrückt halten. Operationen, die den Rechner-Status ändern, werden auf den Seiten 30 (Komplement-Modi), 32 (Wortlänge) und 36 (Flags) beschrieben.

[STATUS]-Anzeige im Einschaltzustand †

2 - 16 - 0000

Komplement-Modus

Flag-Indikatoren

Wortlänge (dezimal)

* Die Flagnummer ist dezimal einzugeben. Beachten Sie, daß die Flagnummer nicht in der Anzeige erscheint.

** Beim ersten Einschalten und nach einem Löschen des Permanentspeichers.

Komplement-Status. Der Komplement-Modus wird durch eine **0** (Unsigned), eine **1** (Einerkomplement) oder eine **2** (Zweierkomplement) dargestellt.

Flagzustände. Der Zustand der Flags 0 bis 3 wird durch vier rechts in der Anzeige erscheinende Ziffern (von rechts nach links) angedeutet. Eine 1 besagt, daß der entsprechende Flag gesetzt ist. Betrachten Sie zum Beispiel die folgende Anzeige des Flag-Status:

#3210 -0100	Flag 2 gesetzt	#3210 -1101	Flags 0, 2 und 3 gesetzt

Sonder-Anzeigen

Statusanzeigen

Der HP-16C enthält sechs Status-Anzeigen, die den Zustand des Rechners in Zusammenhang mit verschiedenen speziellen Operationen aufzeigen. Die Bedeutung und Verwendung dieser Statusanzeigen werden auf den folgenden Seiten erläutert:

*	Spannungsabfallanzeige; Seite 38.
f und g	Vorwahl für Alternativfunktionen; Seite 17.
C	Flag 4 (Carry) gesetzt; Seite 39.
G	Flag 5 (Out-Of-Range) gesetzt; Seite 40.
PRGM	Programm-Modus; Seite 72

Fehler-Anzeige

Bei dem Versuch eine unzulässige Operation auszuführen (z.B. Spezifikation einer Wortlänge größer als 64) zeigt der Rechner eine Fehlermeldung an. In Anhang A finden Sie eine vollständige Auflistung der Fehlermeldungen und ihrer Ursachen.

Sie können die Fehlermeldung durch Drücken einer beliebigen Taste löschen und anschließend Ihre Berechnungen fortsetzen.

Spannungsabfallanzeige

Ein links unten in der Anzeige blinkender Stern signalisiert eine abfallende Betriebsspannung. Die zur Verfügung stehende Restspannung reicht danach jedoch immer noch aus, um ein kontinuierlich laufendes Programm 15 Minuten weiter zu betreiben, oder um mindestens für eine Stunde weitere manuelle Berechnungen durchzuführen. (Bestimmte Batterien haben noch mehr Reserven.) In Anhang C wird das Auswechseln der Batterien beschrieben.

Abschnitt 4

Arithmetische Funktionen und Bitmanipulationen

Ganzzahlige arithmetische Operationen und Bitmanipulationen können nur im Integer-Modus ausgeführt werden. Da bei diesen Funktionen Carry und Out-Of-Range Bedingungen auftreten können, werden wir diese Bedingungen hier kurz erläutern, bevor wir die Funktionen selber besprechen.*

Arithmetische und andere Operationen im dezimalen Gleitkomma-Modus werden in Abschnitt 5, Gleitkommazahlen, beschrieben.

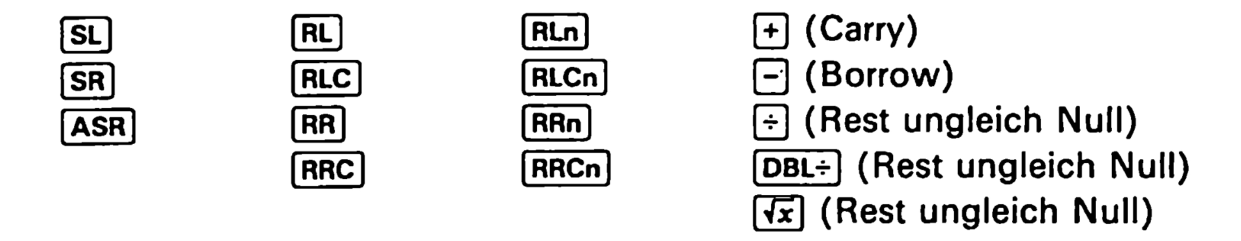
Carry und Out-Of-Range Bedingungen

Bei bestimmten arithmetischen Operationen und Bitmanipulationen können *Carry* und *Out-Of-Range* (Bereichsüberschreitung) Bedingungen auftreten. Diese Zustände setzen Flags (die abgefragt werden können) und Status anzeigen. Die Definition von «Carry» und «Out-Of-Range» Bedingungen hängt von der jeweils ausgeführten Funktion ab.

In Abschnitt 3 (siehe Flags, Seite 36) wird das manuelle Setzen und Löschen dieser (und anderer) Flags beschrieben.

Flag 4: Carry (C)

Die Ausführung der im folgenden gelisteten Shift-, Rotate- und arithmetischen Funktionen bewirkt das Setzen *oder* Löschen von Flag 4 und der Statusanzeige C. Flag 4, der Carry-Flag, wird gesetzt, wenn das Carry-Bit den Wert 1 annimmt, und gelöscht, wenn das Carry-Bit den Wert 0 annimmt.



(Diese genannten Funktionen werden später innerhalb dieses Abschnitts erläutert.)

* Der Anhang A enthält eine Tabelle der relevanten Funktionen und ihrer Wirkung auf den Carry und Out-Of-Range Flag.

Beispiel: Die folgenden einfachen Additionen setzen den Carry Flag (Flag 4) und löschen ihn wieder.

Tastenfolge	**Anzeige**	([STATUS]:2-16-0000)*
[HEX] FFFF [ENTER]	FFFF h	Hexadezimal-Modus.
1[+]	0 h	**C** Statusanzeige. Ein Übertrag ist erfolgt und Flag 4 gesetzt.
1[+]	1 h	Carry-Flag ist gelöscht, da kein neuer Übertrag erfolgt ist.

Flag 5: Out-Of-Range (G)

Flag 5 und die Statusanzeige **G** werden gesetzt, wenn das Ergebnis einer Operation in der derzeitigen Einstellung Komplementmodus nicht dargestellt werden kann. Für die Operationen [+] und [-] entspricht das einer Überlauf (*Overflow*)-Bedingung bei anderen Rechnern.

Folgende Funktionen setzen oder löschen Flag 5 und die Statusanzeige **G**, wenn sie im Integer-Modus ausgeführt werden:



Die arithmetischen Operationen [+], [-], [×] und [÷] beeinflussen Flag 5 auch im dezimalen Gleitkomma-Modus. Zusätzlich gilt dies auch für die Funktion [FLOAT] (siehe Abschnitt 5).

Wird bei einer Operation der Zahlenbereich überschritten, erhalten Sie als Ergebnis nur die letzten Stellen (soviel die gegebene Wortlänge erlaubt) des vollen Ergebnisses. Bei Multiplikation und Division im Einer- oder Zweierkomplement-Modus zeigt das Vorzeichenbit das richtige Vorzeichen des vollen Ergebnises.

---

* In diesem Handbuch wird durchgehend diese Statusanzeige benutzt, um aufzuzeigen, welcher Rechner-Status dem darauffolgenden Beispiel zugrundeliegt.
** Bewirkt immer ein Löschen von Flag 5.

Tastenfolge	Anzeige	([STATUS]:2-16-0000)
[DEC] 32767 [ENTER]	**32767 d**	
2 [×]	**32766 d**	Überlauf setzt Statusanzeige **G** und Flag 5. Führende Binärstelle ist Null; die Zahl ist positiv.
[g] [CF] 5	**32766 d**	Löscht Flag 5.

Flag 5 kann während des Ablaufs eines Programms gesetzt werden; die Programmausführung wird dadurch nicht angehalten.

Arithmetische Funktionen

Die Grundrechenarten

Die arithmetischen Operationen [+], [−], [×] und [÷] können mit ganzzahligen Werten in allen vier Zahlensystemen durchgeführt werden. Die Operanden, die auch in verschiedenen Zahlensystemen eingegeben werden können, müssen sich im X- und Y-Register befinden. Nach Durchführung der Operation fällt der Stack, und das Ergebnis steht im X-Register.

Im Integer-Modus bedingt [÷] eine ganzzahlige Division, der gebrochene Teil des Quotienten wird unterdrückt. Alle arithmetischen Operatoren setzen oder löschen die Flags 4 und 5, mit Ausnahme von [×], der nur auf Flag 5 wirkt.

Beispiel: Berechnen Sie $(5A0_{16})$: (17764_{8})

Tastenfolge	Anzeige	([STATUS]:2-16-0000)
[HEX] 5A0 [ENTER]	**5A0 h**	Eingabe der ersten Zahl.
[OCT] 177764	**177764 o**	Oktal-Modus; Eingabe der zweiten Zahl.
[÷]	**177610 o**	Das oktale Ergebnis ist exakt, da kein Übertrag (Carry) aufgetreten ist.
[HEX]	**FF88 h**	Umwandlung in Hexadezimalbasis.

Addition und Subtraktion im Einerkomplement-Modus. Im Zweierkomplement- und Unsigned-Modus ist das Ergebnis einer Addition oder Subtraktion einfach die Summe oder Differenz der zwei Bitmuster im X- und Y-Register.

Im Einerkomplement-Modus wird dagegen das Ergebnis einer Addition durch das Auftreten eines Übertrags (*Carry*), das Ergebnis einer Subtraktion durch das Auftreten einer «Anleihe» (*Borrow*) beeinflusst. Tritt ein Übertrag auf, wird zum Ergebnis der Wert 1 hinzuaddiert. Bei einem Borrow in das höchstwertige Bit wird das Ergebnis um den Wert 1 subtrahiert. In beiden Fällen wird Flag 4 gesetzt.

(STATUS:1-04-1000)

Übertrag

```
            111
  -1        1110
+(-1)      +1110
-----      -----
 -2₁₀       1100
           +   1
           -----
           1101₂
```

Kein Übertrag

```
 -3         1100
+ 3        +0011
----       -----
 -0₁₀      1111₂
```

Anleihe

```
            1
  3        ø011
 -4       -0100
----      -----
 -1₁₀      1111
          -   1
          -----
          1110₂
```

Keine Anleihe

```
             0 1
  6         01x0
 -5        -0101
----       -----
  1₁₀      0001₂
```

Der Carry-Flag bei Additionen. Der Carry-Flag (Flag 4, Statusanzeige **C**) wird immer gesetzt, wenn sich in einer binären Addition ein Übertrag aus dem höchstwertigen Bit ergibt. Andernfalls wird der Carry-Flag gelöscht. Dies gilt für alle Komplement-Modi.

(STATUS:2-04-1000)

Carry Flag wird gesetzt

```
            ¹1010
  -6
+(-4)       +1100
-----       -----
  6₁₀       0110₂
```

Carry Flag wird gelöscht.

```
  6          0110
 +1         +0001
----        -----
  7₁₀       0111₂
```

(Falsches Ergebnis aufgrund der Wertlänge; daher wird zusätzlich der Out-Of-Range Flag gesetzt.)

Der Carry-Flag bei Subtraktionen. Der Carry-Flag (Flag 4, Statusanzeige **C**) wird immer gesetzt, wenn sich in einer binären Subtraktion eine Anleihe in das höchstwertige Bit ergibt. Andernfalls wird der Carry-Flag gelöscht. Dies gilt für alle Komplement-Modi. (Beachten Sie, daß eine Subtraktion im HP-16C nicht als Addition der entsprechenden negativen Zahl ausgeführt wird; dies beeinflusst auch die Entstehung eines Übertrags).

([STATUS]:2-04-1000)

Carry Flag wird gesetzt		Carry Flag wird gelöscht	
	0 1		0 1
−6	~~1~~010	6	01~~1~~0
−(−4)	−1100	− 1	−0001
-2_{10}	1110_2	5_{10}	0101_2

Der Out-Of-Range Flag. Wenn ein arithmetisches Ergebnis in der momentanen Wortlänge und Komplement-Modus nicht dargestellt werden kann, wird der Out-Of-Range Flag gesetzt. Bei Divisionen kann dies nur im Zweierkomplement-Modus vorkommen, wenn die größtmögliche negative Zahl durch −1 geteilt wird.

Beispiel: Berechnen Sie (7 + 6) im Binär-Modus bei einer Wortlänge von 4 Bit und beachten Sie die Auswirkungen auf Flag 4 und 5.

Tastenfolge	Anzeige	([STATUS]:2-04-0000)
[BIN]		Binär-Modus.
111 [ENTER]	**111 b**	7.
110	**110 b**	6.
[+]	**1101 b**	−3. Flag 5 (Out-Of-Range) gesetzt, Flag 4 (Carry) gelöscht.

Divisions-Rest und [RMD]

Bei einer Divison wird nur der ganzzahlige Anteil des Divisionsergebnisses in das X-Register übertragen. Ist der Rest ungleich Null, werden Flag 4 (Carry) und die Statusanzeige **C** gesetzt. Verbleibt kein Rest, wird Flag 4 gelöscht. Um den Rest anstelle des Quotienten zu erhalten, drücken Sie [f] [RMD] anstelle von [÷]. Diese Funktion berechnet $|y|$ *MOD* $|x|$, wobei das Vorzeichen des Ergebnisses entsprechend dem Vorzeichen des Dividenden y gesetzt wird.

Tastenfolge	**Anzeige**	([STATUS]:2-16-0000)
[HEX] 66 [ENTER]	**66 h**	Hexadezimal-Modus.
7 [÷]	**E h**	Statusanzeige **C**. 7 teilt 66 mit Rest.
2 [÷]	**7 h**	Keine Statusanzeige, Flag 4 gelöscht. 2 teilt E ohne Rest.
4 [f] [RMD]	**3 h**	Rest von 7/4.

Quadratwurzel

Die Funktion [√x] berechnet die Quadratwurzel der Zahl im X-Register. Weist das Ergebnis Stellen nach dem Komma auf, werden diese unterdrückt und Flag 4 (Carry) gesetzt; andernfalls wird Flag 4 gelöscht.

Negative Zahlen und Komplementbildung

Vorzeichenwechsel. Die Funktion [CHS] (*change sign*) wechselt das Vorzeichen, indem sie das (Einer- bzw. Zweier-) Komplement der Zahl im X-Register bildet. Befindet sich im X-Register die größtmögliche negative Zahl im Zweierkomplement-Modus, dann wird nur Flag 5 (Out-Of-Range) gesetzt, die Zahl selbst wird nicht verändert.

Im Unsigned-Modus bewirkt [CHS] die Bildung des Zweierkomplements. Flag 5 (Statusanzeige **G**) wird gesetzt als Hinweis, daß negative Zahlen den Wertebereich des Unsigned-Modus überschreiten.

Um eine negative Zahl einzugeben, drücken Sie [CHS] *nach* der Eingabe der Ziffern. Im Integer-Modus beendet [CHS] die Ziffereingabe.

Absolutwert. Die Funktion [g] [ABS] ermittelt den Betrag der Zahl im X-Register, indem sie das Einer- bzw. Zweierkomplement bei einer negativen Zahl berechnet. Positive Zahlen und Zahlen im Unsigned-Modus werden nicht verändert.

Befindet sich im X-Register die größtmögliche negative Zahl im Zweierkomplement-Modus, so bewirkt [ABS] nur das Setzen von Flag 5 (Out-Of-Range).

Logische Operationen

Die logischen (*booleschen*) Operationen NOT, OR, AND und EXCLUSIVE OR liefern das Ergebnis einer bitweisen Analyse einer bzw. zweier Binärzahlen.

Die Operatoren [OR], [AND] und [XOR] arbeiten mit den Bits entsprechender Position der Worte im X- und im Y-Register, das Ergebnis wird im X-Register abgelegt und der Stack fällt. Der Operator [NOT] wirkt nur auf das Wort im X-Register, der Stack fällt nicht.

NOT

Die Funktion [NOT] invertiert die Werte aller Bits der Binärzahl im X-Register. Dies ist äquivalent mit der Bildung des Einerkomplements, d.h. mit der Wirkung von [CHS] im Einerkomplement-Modus. Nur der Inhalt des X-Registers wird verwendet.

Tastenfolge	**Anzeige**	([STATUS]:2-16-1000)
[BIN] 11111111	**11111111 b**	Binär-Modus.
[f] [NOT]	**00000000 .b**	Das Einerkomplement zu
[f] [WINDOW] 1	**11111111 b.**	0000000011111111_2 ist 1111111100000000_2.

AND

Die Funktion [AND] (*logisches Produkt*) vergleicht die einander entsprechenden Bits zweier Worte. Jedes Ergebnisbit hat immer genau dann 1, wenn die beiden entsprechenden Operandenbits den Wert 1 haben und ansonsten 0. Ein Beispiel für die Benutzung von [AND] finden Sie unter «Masken» auf Seite 51.

OR

Die Funktion [OR] (*logische Summe*) vergleicht die einander entsprechenden Bits zweier Worte. Jedes Ergebnisbit ist genau dann 0, wenn beide Eingabebits der entsprechenden Stelle den Wert 0 haben und ansonsten 1.

Beispiel: Stellen Sie mit Hilfe einer logischen ODER-Operation fest, welche Bits in 10101_2 und 10011_2 *gleichzeitig* 0 sind.

Tastenfolge	**Anzeige**	([STATUS]:2-16-0000)
10101 [ENTER]	**10101 b**	
10011	**10011 b**	
[f] [OR]	**10111 b**	Bit 3 (durch die 0 dargestellt) und alle Bits links von Bit 4 haben in beiden Worten den Wert 0.

EXCLUSIVE OR

Die Funktion [XOR] (*logische Differenz*) vergleicht die entsprechenden Bits zweier Worte. Das resultierende Ergebnisbit wird nur dann auf 1 gesetzt, wenn die beiden verglichenen Bits unterschiedliche Werte haben.

Beispiel: Stellen Sie mit Hilfe der Funktion [XOR] fest, ob die beiden Binärzahlen 01010101_2 und 01011101_2 gleich sind. Eine 1 im Ergebnis deutet an, daß sich die beiden verglichenen Werte in dieser Bitposition unterscheiden.

Tastenfolge	Anzeige	([STATUS]:2-16-0000)
1010101 [ENTER]	**1010101 b**	
1011101	**1011101 b**	
[f] [XOR]	**1000 b**	Die beiden Binärzahlen unterscheiden sich in der vierten Binärposition von rechts.

Shift- und Rotate-Operationen

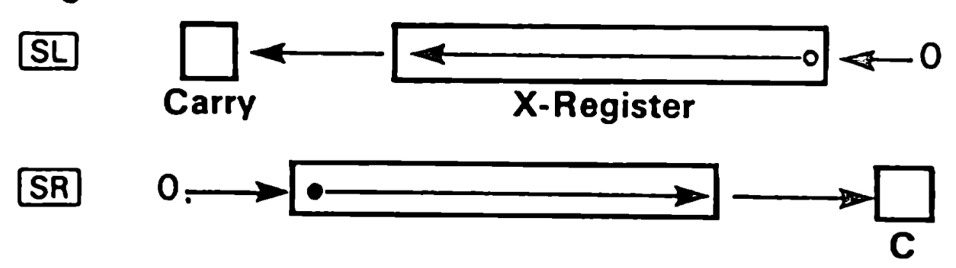
Shift- und Rotate-Operationen bewegen die Bits eines Wortes nach rechts oder links. Die spezifische Operation legt jeweils den Wert des am einen Wortende neu eintretenden Bits und die Vewendung des am anderen Ende herausfallenden Bits fest.

Flag 4 (Carry) wird von jeder Shift- und Rotate-Operation mit Ausnahme von [LJ] (*left justify*) gesetzt oder gelöscht, wie die nachfolgenden Diagramme zeigen.

Shift-Operationen

Die Shift-Operationen lassen sich in *logische* Shifts und *arithmetische* Shifts unterteilen. Bei arithmetischen Shifts bleibt das Vorzeichenbit erhalten. Des weiteren können Sie mittels einer speziellen Shift-Operation den Inhalt des X-Registers *linksbündig ausrichten.*

Logische Shifts. [f] [SL] bzw. [f] [SR] verschiebt alle Bits des Worts im X-Register um eine Stelle nach links bzw. nach rechts. Aus dem Wort herausgeschobene Bits werden im Carry-Bit abgelegt, dessen vorhergehender Zustand dabei verloren geht. Die am anderen Wortende freiwerdenden Bitpositionen werden immer auf 0 gesetzt.

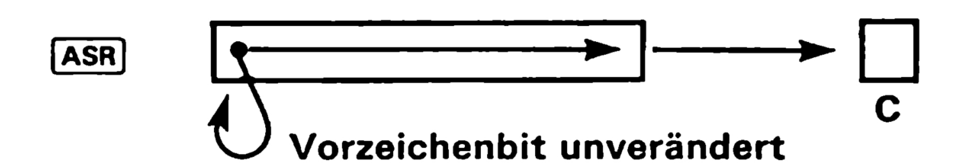


Linksbündige Ausrichtung. Um ein Bitmuster innerhalb einer gegebenen Wortlänge linksbündig auszurichten, drücken Sie [g] [LJ]. Es erfolgt ein Stack Lift, das linksbündige Wort steht im Y-Register, und das X-Register enthält die Anzahl der benötigten Bit-Shifts. [LJ] beeinflußt nicht den Zustand des Carry-Flag.

Beispiel: Das Bitmuster 1111 soll bei einer Wortlänge von 8 linksbündig ausgerichtet werden.

Tastenfolge	Anzeige	([STATUS]:2-08-0000)
1111	1111 b	
[g] [LJ]	100 b	Anzeige des Zählers; zur Ausrichtung wurden vier Bit-Shifts benötigt.
[R↓]	11110000 b	Linksbündiges Wort.

Arithmetische Rechts-Shifts. Die Tastenfolge [g] [ASR] (*arithmetic shift right*) verschiebt wie [SR] das Wort im X-Register um eine Stelle nach rechts. Jedoch wird die links entstehende Lücke nicht mit einer Null, sondern mit dem Wert des Vorzeichenbits aufgefüllt. Im Unsigned-Modus (kein Vorzeichenbit) ist [ASR] mit [SR] identisch. Das Carry-Bit wird entsprechend der Wertigkeit des aus dem X-Register herausgeschobenen Bits gesetzt.



Beispiel: Die Verschiebung einer positiven Binärzahl um *n* Stellen nach rechts ist äquivalent zu einer Division der Zahl durch 2^n. Da das Vorzeichenbit erhalten bleibt, kann die arithmetische Verschiebung auch zur Division negativer Zahlen durch 2 benutzt werden*. Dividieren Sie erst 0111111 und dann 1000000 durch 2^3 (Wortlänge 8).

Tastenfolge	Anzeige	([STATUS]:2-08-0000)
[g] [SF] 3		Führende Nullen werden angezeigt.
1111111	0111111 b	

* Bei *ungeraden* negativen Zahlen im Zweierkomplement-Modus ergibt [ASR] ein um den Wert 1 kleineres Ergebnis als eine Division durch 2.

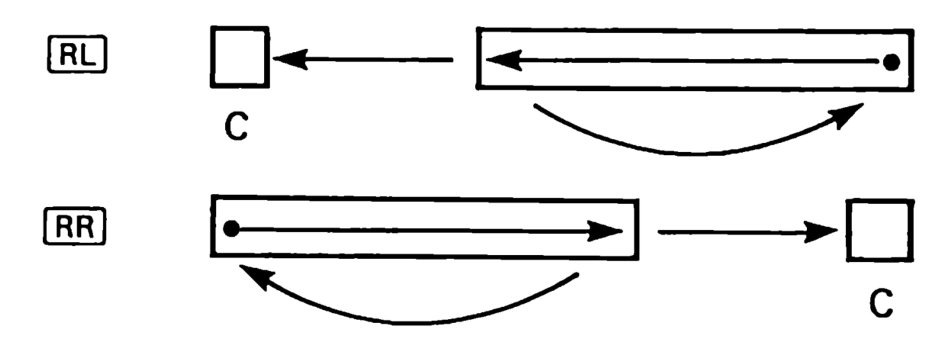
Tastenfolge	Anzeige	([STATUS]:2-08-0000)
[f] SHOW [DEC]	127 d	[ENTER] wird nicht benötigt, da SHOW die Ziffereingabe beendet.
[f][SR] [f][SR] [f][SR]	00001111 b	Jeder Shift bewirkt eine *ganzzahlige* Division durch 2 und setzt Flag 4, da eine 1 in das Carry-Bit geschoben wird.
[f] SHOW [DEC] (gehalten)	15 d	
(losgelassen)	00001111 b	
10000000	10000000 b	
[f] SHOW [DEC]	–128 d	
[g] [ASR]	11000000 b	
[g] [ASR]	11100000 b	
[g] [ASR]	11110000 b	Jeder Shift dupliziert das Vorzeichenbit und löscht den Carry-Flag.
[f] SHOW [DEC]	–16 d	
(losgelasssen)	11110000 b	

Rotate-Funktionen

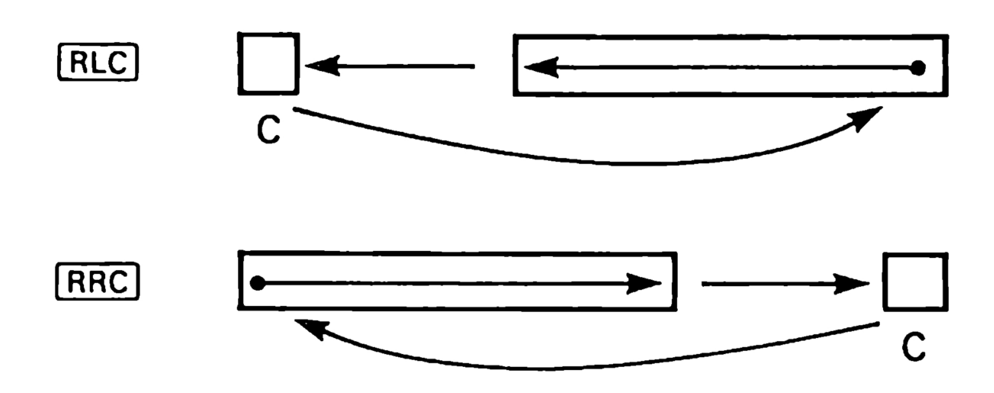
Die Rotate-Funktionen des HP-16C lassen sich in drei Klassen mit insgesamt acht Funktionen einteilen.

- Rundumverschiebung nach links und nach rechts ([RL], [RR]).
- Rundumverschiebung nach links/rechts «durch das Carrybit» ([RLC], [RRC]).
- Rundumverschiebung um *n* Stellen ([RLn], [RRn], [RLCn], [RRCn]).

Einfache Rundumverschiebungen. Die Tastenfolgen [f] [RL] (*rotate left*) und [f] [RR] (*rotate right*) bewirken eine Rundumverschiebung des Worts im X-Register um ein Bit nach links bzw. rechts. Aus dem Wort herausgeschobene Bits treten am anderen Ende wieder ein. Der Carry-Flag wird gesetzt, wenn das herausgeschobene Bit den Wert 1 hat, und andernfalls gelöscht.



Rundumverschiebungen durch das Carrybit. Die Funktionen [RLC] (*rotate left through carry*) und [RRC] (*rotate right through carry*) laden bei der Rundumverschiebung das herausgeschobene Bit in das Carry-Bit und den Inhalt des Carry-Bits an das andere Wortende.



Rundumverschiebungen um mehrere Stellen. Ist das Y-Register mit einem Bitmuster und das X-Register mit einer Zahl *n* geladen, dann bewirken [f] [RLn], [f] [RRn], [g] [RLCn] und [g] [RRCn] eine Rundumverschiebung des Bitmusters um |*n*| Stellen. Der Stack fällt und das Ergebnis wird im X-Register abgelegt.

Der Carry-Flag hat den gleichen Status, wie nach einer |*n*|-maligen Hintereinanderausführung von [RL], [RR], [RLC] oder [RRC]. Zum Beispiel ist der Carry-Flag nach [RRn] mit *n* = 3 nur dann gesetzt, wenn Bit 2 (das dritte Bit von rechts) 1 ist.

Beispiel: gesucht ist eine Tastenfolge, die ein *in zwei 8-Bit Worte aufgeteiltes* und in zwei verschiedenen Registern gespeichertes 16-Bit Wort als *ganzes* Wort nach links rundumverschiebt. Testen Sie die Tastenfolge bei einer Wortlänge von 8 anhand des Worts 0001110011100111.

Tastenfolge	**Anzeige**	([STATUS]:2-08-1000)
11100	**00011100 b**	Höherwertiger Teil des 16-Bit Worts.
[f] [SL]	**00111000 b**	Schiebt das höchstwertige Bit ins Carry-Bit.
[g] [LSTx]	**00011100 b**	Zurückrufen des höherwertigen Teils.
11100111	**11100111 b**	Niederwertiger Teil des 16-Bit Worts.

Tastenfolge	Anzeige	([STATUS]:2-08-1000)
[g] [RLC]	11001110 b	Verschiebt das Carry-Bit (höchstwertiges Bit der ersten 8 Stellen des Worts) in das niedrigstwertige Bit der letzten 8 Stellen des Worts.
[x≷y]	00011100 b	Vertauscht X- und Y-Register.
[g] [RLC]	00111001 b	C gelöscht: das Carrybit wandert in den ersten Teil des Worts und wird durch den Wert 0 ersetzt.
[x≷y]	11001110 b	Das neue Wort ist 00111001 11001110.
[g] [CF] 3	11001110 b	Unterdrückt führende Nullen.

Setzen, Löschen und Abfragen von Bits

Einzelne Bits eines Wortes können mit den Funktionen [SB] (*set bit*) und [CB] (*clear bit*) auf den Wert 1 gesetzt oder auf den Wert 0 gelöscht werden. Ähnlich wie bei Flags können Sie den Wert einzelner Bits mit [B?] abfragen. Bei der Ausführung innerhalb eines Programms kann das Ergebnis zur Programmsteuerung benutzt werden.

Um ein bestimmtes Bit eines Wortes zu setzen, zu löschen oder abzufragen:

- Laden Sie das entsprechende Wort in das Y-Register.
- Laden Sie die Nummer des zu bearbeitenden Bits in das X-Register ein.

Nach dem Drücken von [SB] oder [CB] erfolgt Stack Drop, und das bearbeitete Wort steht wieder im X-Register.

Den Bits eines Wortes sind bei einer Wortlänge von *n* von rechts nach links die Nummern 0 bis *n*-1 zugeordnet.

Tastenfolge	Anzeige	([STATUS]:2-16-0000)
11111111 [ENTER]	11111111 b	Eingabe des Wortes ins Y-Register.
11	11 b	Bit Nummer 3.
[f] [CB]	11110111 b	Stack fällt; Ergebnis steht im X-Register.

Die Abfrage eines bestimmten Bits mit [f] [B?] wird in der Programmierung häufig bei *bedingten Verzweigungen* verwendet; eine Entscheidung im Programmablauf kann vom Bitmuster einer Zahl abhängig gemacht werden. (Die X- und Y-Register müssen dabei – wie zuvor beschrieben – die gewünschten Werte enthalten). Bedingte Verzweigungen werden in Abschnitt 9 behandelt.

Masken

Die Funktionen [MASKL] (*mask left*) und [MASKR] (*mask right*) erzeugen rechts- oder linksbündige Masken, d.h. Binärzahlen einer vorgegebenen Länge, die nur aus Einsen bestehen. Die Zahl im X-Register bestimmt dabei die Länge der Maske. Nach Ausführung der Funktion steht die Maske im X-Register (der Stack fällt nicht).

Sie können Masken bis zur vollen Wortlänge erzeugen. Um eine Maske in die Mitte eines Wortes zu verschieben, können Sie [MASKL] oder [MASKR] zusammen mit einer Shift-Funktion benutzen.

Beispiel: Die ASCII-Darstellung einer zweistelligen Zahl belegt 16 Bit, – acht Bit pro Ziffer. Gegeben sei die ASCII-Zahl 65 (00110110 00110101); extrahieren Sie die höherwertige Ziffer, indem Sie die Hälfte dieses ASCII-Codes in den äquivalenten binär kodierten Dezimalwert (BCD-Wert) umwandeln.

	0011 0110 0011 0101 ASCII«65» («3», «6», «3», «5»)
[AND]	0000 1111 0000 0000 Maske
	0000 0110 0000 0000 extrahierte Sie die höherwertige Ziffer («6»)

Sie können in diesem Beispiel Tastenfolgen einsparen, wenn Sie die Ziffern der Zahl vor der Maskierung positionieren.

Tastenfolge	Anzeige	([STATUS]:2-16-0000)
[HEX] 3635 [ENTER]	3635 h	
8 [f] [RRn]	3536 h	Rundumverschiebung des Worts um 8 Bit nach rechts; die gesuchte Hexadezimal-Ziffer («6») ist jetzt rechtsbündig.
4 [f] [MASKR]	F h	Erzeugt eine rechtsbündige Maske aus vier 1er Bits (1111) innerhalb eines 16-Bit Worts.
[f] [AND]	6 h	Extrahiert die vier am weitesten rechts stehenden Bits («6»).

Summation von Bits

Die Tastenfolge [g] [#B] (*number of bits*) summiert alle Bits des X-Registers und lädt das Ergebnis in das X-Register. Das Bitmuster ist danach im LAST X-Register noch verfügbar. Der Stack wird nicht angehoben. (Bei Wortlänge 1 und 2 müssen die Ergebnisse im Unsigned-Modus interpretiert werden.)

«Doppeltgenaue» Funktionen

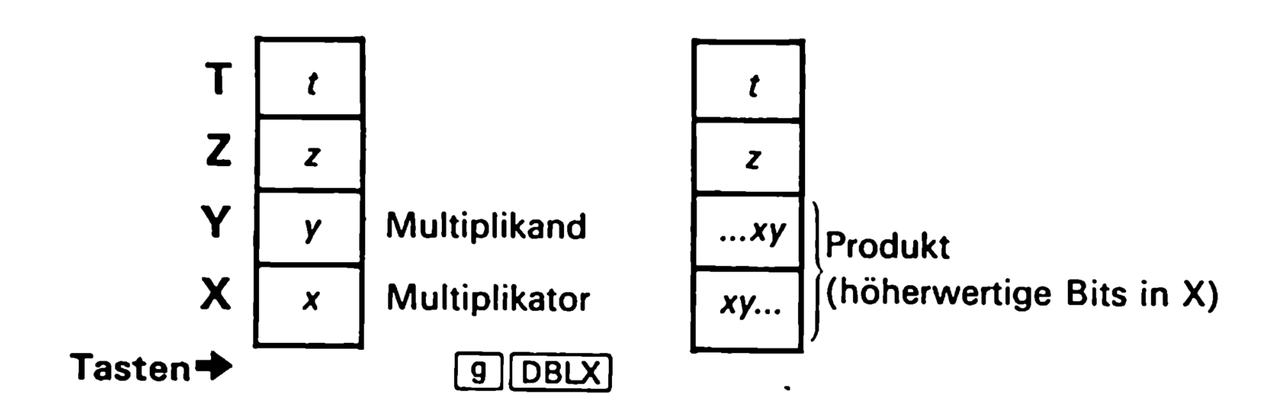
Der HP-16C verfügt über mehrere «doppeltgenaue» Funktionen: [DBLX] (*double multiply*), [DBL÷] (*double divide*) und [DBLR] (*double remainder*). Diese Funktionen erlauben die *exakte* Berechnung eines Produkts doppelter Wortlänge und die *exakte* Berechnung des Quotienten und des Rests für einen Dividenden doppelter Wortlänge.

Um im Hexadezimal- und Oktal-Modus sinnvolle doppeltgenaue Ergebnisse zu erhalten, darf die Wortgrenze (die von der Anzahl der verfügbaren Bits abhängt) eine Ziffernposition nicht aufspalten. Sie sollten deshalb eine passende Wortlänge wählen: ein Vielfaches von 4 im Hexadezimal-Modus, ein Vielfaches von 3 im Oktal-Modus.

Doppeltgenaue Multiplikation

Die Funktion [DBLX] multipliziert zwei Größen einfacher Wortlänge in den Registern X und Y zu einem Doppelwort-Ergebnis in den Registern X und Y (kein Stack Drop). Das Ergebnis ist rechtsbündig ausgerichtet; das X-Register enthält die *höherwertigen*, das Y-Register die *niederwertigen* Stellen des Ergebnisses.

Die folgende Illustration zeigt die Inhalte des Stack während dieser Operation. Der Stack ist mit den Werten *t*, *z*, *y* und *x* geladen, jedes Register enthält ein Wort.



* Abschnitt 7, Grundlagen der Programmierung, enthält ein Programm zur Benutzung von [DBLX] im Dezimal-Modus (siehe Seite 78).

Beispiel: Die folgende Berechnung von (7 × 6) mit Wortlänge 5 und Zweierkomplement veranschaulicht die Verwendung des Doppelprodukts.

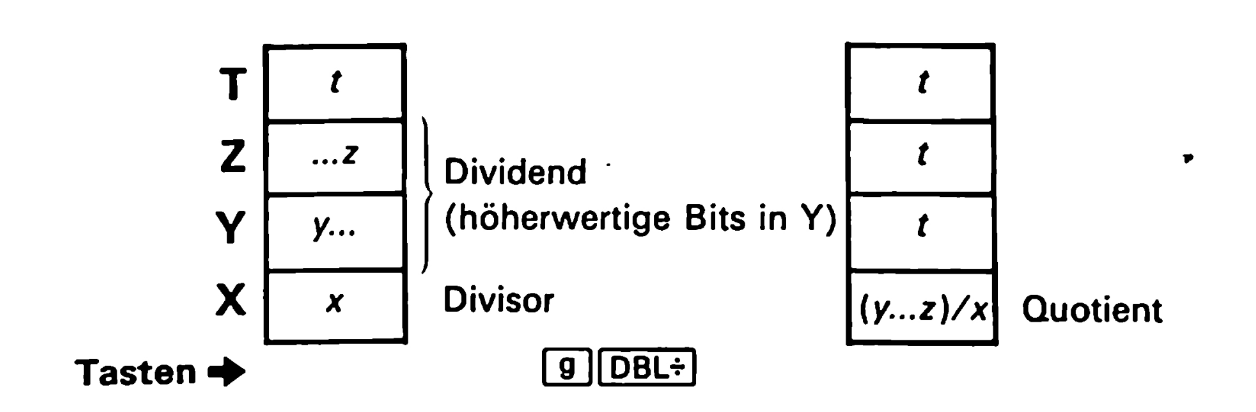
7	00111	Fünf Bits in Y
× 6	× 00110	Fünf Bits in X
42_{10}	00001 01010_2 (X: 00001, Y: 01010)	10-Bit Darstellung von 42_{10}, aufgeteilt zwischen X- und Y-Register.

Tastenfolge	Anzeige	([STATUS]:2-05-1000)
[BIN] 111 [ENTER]	00111 b	Binär-Modus.
110 [g] [DBLX]	00001 b	Höherwertige Bits in X.
[x⇄y]	01010 b	Niederwertige Bits in Y. Das Ergebnis ist daher 00001 01010_2.

Doppeltgenaue Division

Die Funktion [DBL÷] berechnet den Quotienten eines Dividenden doppelter Wortlänge in den Registern Y und Z, und eines Divisors einfacher Wortlänge im X-Register. Der Stack fällt zweimal; das Ergebnis steht danach im X-Register.

Die Meldung **Error 0** wird angezeigt, wenn das Ergebnis nicht in einfacher Wortlänge dargestellt werden kann. Flag 4 (Carry) wird gesetzt, wenn der Rest ungleich Null ist. Der Stack ist während dieser Operation wie folgt belegt:



Beispiel: Die folgende Illustration zeigt die binäre Berechnung von (–88 ÷ 11) im Zweierkomplement-Modus bei einer Wortlänge von 5.

```
                                X
     - 8₁₀      X             11000₂     5-Bit Ergebnis in X.
 11 | -88    01011 | 11101  01000        10-Bit Darstellung von -8₁₀, auf
                      Y       Z          Y- und Z-Register verteilt.
```

Tastenfolge	Anzeige	([STATUS]:2-05-1000)
1000 [ENTER]	**01000 b**	Niederwertige Bits der 10-Bit Dividenden für das Z-Register.
11101 [ENTER]	**11101 b**	Höherwertige Bits des 10-Bit Dividenden für das Y-Register.
1011 [g] [DBL÷]	**11000 b**	Quotient.
[g] [CF] 3	**11000 b**	Unterdrückung führender Nullen.

Doppeltgenauer Rest

Die Funktion [DBLR] arbeitet wie [DBL÷], zeigt aber nicht den Quotienten, sondern den Rest an. Überschreitet der *Quotient* 64 Bit, zeigt der Rechner **Error 0** an.

Der Rest wird wie bei der Funktion [RMD] ermittelt (Seite 43); das Ergebnis erhält das Vorzeichen des Dividenden.

Beispiel: Anwendung der doppeltgenauen Division

Berechnen Sie den Quotienten $\frac{5714AF2_{16}}{7E14684_{16}}$ auf 16 Hexadezimalstellen genau. Obwohl das Ergebnis nicht ganzzahlig ist, kann dieses Problem im Integer-Modus gelöst werden: Berechnen Sie zuerst den Quotienten von

$$\frac{5714AF2\overbrace{000...0}^{\text{16 Nullen}}{}_{16}}{7E14684_{16}}$$

und setzen Sie dann einen Dezimalpunkt vor die erste Stelle des Ergebnisses (dies entspricht einer Division des Ergebnisses durch 2^{64}). Benutzen Sie [DBL÷], um einen derart großen Zähler zu verarbeiten.

Tastenfolge	Anzeige	([STATUS]:2-64-0000)
[HEX]		Hexadezimal-Modus.
[f] SET COMPL [UNSGN]		Unsigned Modus hat den größten Wertebereich und verhindert daher eine Bereichsüberschreitung.
0 [ENTER]	**0 h**	Dividend doppelter Länge ist
5714AF2 [ENTER]	**5714AF2 h**	5714AF2 × 2^{64}.
7E14684 [g] [DBL÷]	**7E985d8C .h**	Carry-Bit gesetzt.
[f] [WINDOW] 1	**b0d06F6a h.**	Das Ergebnis lautet: $B0D06F6A7E985D8C_{16}$ die Lösung des ursprünglichen Problem ist also: $0.B0D06F6A7E985D8C_{16}$.

Abschnitt 5

Gleitkomma-Zahlen

Zusätzlich zur ganzzahligen Arithmetik in den verschiedenen Zahlensystemen verfügt der HP-16C über eine dezimale Gleitkomma-Arithmetik. Im dezimalen Gleitkomma-Modus werden die Zahlen linksbündig angezeigt, und die Wortlänge ist automatisch auf 56 Bit eingestellt.

Hinweis: Zahlen werden im Gleitkomma-Modus und im Integer-Modus in zwei verschiedenen, nicht kompatiblen Formaten dargestellt.* In einem Format abgespeicherte Werte werden nicht in der richtigen Weise umgewandelt, wenn Sie den Rechner in das andere Format umschalten. Nach einer Zurückschaltung in das ursprüngliche Format zurückgeschaltet, sind die gespeichertenWerte wieder unverfälscht verfügbar.

Umschalten in den dezimalen Gleitkomma-Modus

Die Funktion [FLOAT] schaltet den Rechner in den dezimalen Gleitkomma-Modus und wandelt die Inhalte des X- und Y-Registers in die entsprechenden dezimalen Gleitkommazahlen um.

Die Tastenfolge [f] [FLOAT] {0 bis 9, [·]} setzt den dezimalen Gleitkomma-Modus. Die gewählte Zahl bestimmt die Anzahl der angezeigten Dezimalstellen; durch Spezifikation [·] wird wissenschaftliche Notation angewählt. Befindet sich der Rechner bereits im Gleitkomma-Modus, wird der Stack nicht weiter umgewandelt.

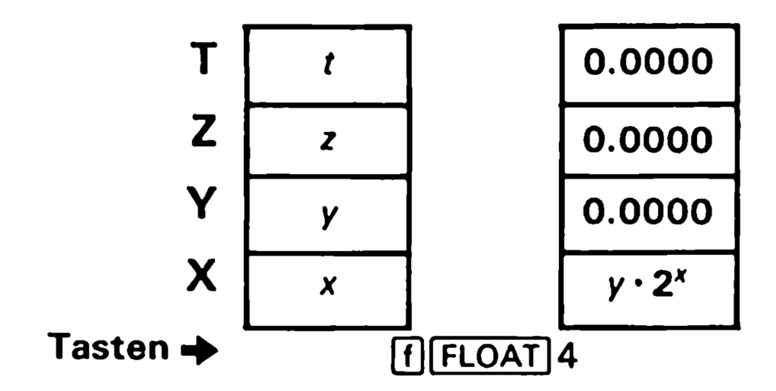
Umwandlung im Stack

Wenn Sie in den Gleitkomma-Modus umschalten, möchten Sie vielleicht eine zuvor abgespeicherte Zahl weiter benutzen. Der HP-16C verfügt über eine Umwandlungs-Routine, die es Ihnen ermöglicht, eine Zahl aus dem Integer-Modus zu speichern und nach dem Umschalten in den Gleitkomma-Modus wieder zu benutzen. Die Funktion [FLOAT] im Integer-Modus wandelt die Zahlen im X- und Y-Register in den Gleitkomma-Dezimalwert des Ausdrucks $(y)(2^x)$ um, der danach im X-Register abgelegt wird. Die übrigen Stackregister werden gelöscht.**

---

* BCD (*Binary Coded Decimal*) im Gleitkomma-Modus, binär im Integer-Modus.

** Anhang D enthält ein Programm zur Umwandlung vom dezimalen HP-16C Gleitkommaformat ins binäre IEEE Gleitkommaformat und zurück.



Ist $y \cdot 2^x$ größer als $9.999999999 \times 10^{99}$, wird Flag 5 (Out-Of-Range) gesetzt, und die Overflow-Anzeige (**9.999999999 99**) erscheint. Ergibt sich kein Overflow, wird Flag 5 gelöscht.

Beispiel: Wandeln Sie $25E47_{16}$ in das dezimale Gleitkomma-Format um.

Tastenfolge	Anzeige	(STATUS:2-64-0000)
HEX		Hexadezimal-Modus.
25E47 ENTER	**25E47 h**	Mantisse.
0	**0 h**	Exponent von 2.
f FLOAT 2	**155,207.00**	Setzt Gleitkomma-Modus mit zwei Nachkommastellen. Die angezeigte Zahl ist äquivalent zu $(25E47_{16}) \cdot 2^0$.

Andere Auswirkungen beim Umschalten in den Gleitkomma-Modus

Das Umschalten vom Integer-Modus (Hexadezimal-, Dezimal-, Oktal- oder Binär-Modus) in den dezimalen Gleitkomma-Modus bewirkt zusätzlich folgende Einstellungen:

- Die Wortlänge wird auf 56 festgelegt.
- Der Stack (außer dem X-Register) und das LAST X Register) werden gelöscht; der Stack Lift wird freigegeben.
- Die Speicherregister werden *nicht* gelöscht. Jedoch führt jeder Zugriffsversuch auf Registerinhalte (einschließlich des Indexregisters), die *nicht* im Gleitkomma-Modus geladen wurden, in der Regel zur Anzeige der Fehlermeldung **Error 6**.*

* Siehe dazu Seite 68.

- Die Funktionen zur Wahl der Komplement-Modi bleiben wirksam, beeinflussen aber *nicht* die arithmetischen Funktionen oder Zahlendarstellungen. Der Komplementmodus *beeinflusst jedoch* die Umwandlung des X-Registers, wenn der Rechner wieder auf Integer-Modus geschaltet wird.

Zifferneingabe und weitere Anzeigeformate

Vorzeichenwechsel. Die Funktion [CHS] ändert das Vorzeichen der Zahl in der Anzeige. Um eine negative Zahl einzugeben, drücken Sie nach der Zifferneingabe [CHS]. Im Gleitkomma-Modus wird damit die Zifferneingabe nicht abgeschlossen.

Wissenschaftliche Notation. Die Tastenfolge [f] [FLOAT] [·] setzt den dezimalen Gleitkomma-Modus *und* schaltet das Anzeigeformat auf wissenschaftliche Notation (Mantisse mit 6 Nachkommastellen + zweistelliger Exponent).

Exponenten. Der Exponent einer Zahl wird mit Hilfe von [EEX] (*enter exponent*) eingegeben. Geben Sie zuerst die Mantisse ein, und drücken Sie danach [f] [EEX], gefolgt vom Exponenten. (Bei negativer Mantisse muß [CHS] vor [EEX] gedrückt werden.) Bei negativem Exponenten drücken Sie [CHS] nach der Eingabe des Exponenten.

Stellen der Mantisse im Anzeigefeld des Exponenten verschwinden aus der Anzeige, wenn Sie [EEX] drücken, bleiben aber intern erhalten.*

Anzeige der Mantisse. Unabhängig vom Anzeigeformat im dezimalen Gleitkomma-Modus wird intern jede Zahl als 10-stellige Mantisse mit zweistelligem Exponenten zur Basis 10 dargestellt. Wollen Sie die gesamte 10-stellige Mantisse einer Zahl im X-Register sehen, müssen Sie [f] CLEAR [PREFIX] drücken. Solange Sie die Taste [PREFIX] gedrückt halten, steht die vollständige Mantisse in der Anzeige.

Tastenfolge	Anzeige	([STATUS]:2-56-0000)
[f] [FLOAT] 3		
45 [g] [√x]	**6.708**	
[f] CLEAR [PREFIX]		
(gehalten)	**6708203932**	
(losgelassen)	**6.708**	

* Um eine mißverständliche Anzeige zu vermeiden, läßt sich [EEX] nicht auf Zahlen mit mehr als sieben Stellen vor dem Dezimalpunkt oder mit einer Mantisse kleiner als 0.000001 anwenden. Zur Eingabe einer derartigen Zahl muß eine Darstellung mit grösserem Exponenten verwendet werden.

Overflow und Underflow. Ist das Ergebnis einer Gleitkommaberechnung eine Zahl mit einer Wertigkeit größer als $9.999999999 \times 10^{99}$, wird anstelle des wahren Ergebnisses der Ersatzwert $\pm 9.999999999 \times 10^{99}$ abgespeichert (die letzten drei Stellen der Mantisse werden allerdings nicht angezeigt). Zusätzlich werden Flag 5 (Out-Of-Range) und die Statusanzeige **G** gesetzt.

Ist das Ergebnis einer Berechnung im Gleitkomma-Modus betragsweise kleiner als $1{,}0 \times 10^{99}$, wird diese Zahl durch Null ersetzt. Overflow und Underflow unterbrechen nicht die Ausführung eines Programms.

Zurückschalten in den Integer-Modus

Wenn Sie eine der Zahlenbasis-Tasten drücken, wird der Rechner automatisch in den Integer-Modus zurückgeschaltet.

Umwandlung im Stack

Wenn Sie den Gleitkomma-Modus verlassen, werden X- und Y-Register in umgekehrter Weise wie beim Umschalten in den Gleitkomma-Modus umgewandelt. Der Inhalt des X-Registers wird dabei als Ausdruck der Form $(y)(2^x)$ interpretiert, wobei der ganzzahlige Wert y im Y-Register und der Exponent (zur Basis 2) x im X-Register abgelegt wird. Der Wert für y (wenn y ungleich 0) ist dabei so definiert, daß $2^{31} \leqslant |y| < 2^{32}$ gilt, d.h. die Mantisse y wird auf eine 32-Bit Ganzzahl gerundet. Der Wert des Exponenten x ist durch y in der Weise festgelegt, daß $(y)(2^x)$ dem Wert des X-Registers vor der Umwandlung entspricht.

Die neuen x- und y-Werte werden im gewählten Zahlensystem ausgedrückt. (Im Unsigned-Modus werden die Absolutwerte von x und y in X und Y abgelegt.)

Die Umwandlung ganzzahliger Werte y und x vom Integer- in den dezimalen Gleitkomma-Modus und zurück wird im allgemeinen nicht mehr das *gleiche*, aber auf jeden Fall ein *äquivalentes* Zahlenpaar ergeben.

Beispiel: Verwandeln Sie 1.284×10^{-17} in ein Produkt einer dezimalen Ganzzahl mit einer Potenz von 2.

Tastenfolge	**Anzeige**		([STATUS]:2-56-0000)
1.284	1.284		
[f] [EEX]	1.284	00	
17 [CHS]	1.284	–17	

Tastenfolge	Anzeige	
[DEC]	–88 d	Schaltet den Rechner in Integer-Modus; das X-Register enthält den Exponenten zur Basis 2.
[x≷y]	73787526 .d	Mantisse.
[f] [WINDOW] 1	39 d.	Das Ergebnis lautet $(3973787526 \times 2^{-88})_{10}$.

Beispiel: Die folgende Tastenfolge zeigt, daß die Umwandlung einer 1 im Integer-Modus $(1 \cdot 2^0 = 1)$ in den Gleitkomma-Modus und zurück $(80000000_{16} \times 2^{-31})$ einen zwar gleichwertigen, aber in der Darstellung verschiedenen Ausdruck ergibt.

Tastenfolge	Anzeige	([STATUS]:2-56-0000)
[HEX]		
1 [ENTER]	1 h	$y = 1$. Es gilt nicht $2^{31} \leq \lvert y \rvert < 2^{32}$.
0	0 h	$x = 0$.
[f] [FLOAT] 4	1.0000	$(y) \cdot (2^x) = (1) \cdot (2^0) = 1$.
[HEX]	FFFFFFE1 .h	Zurückschalten in den Integer-Modus.
[f] SHOW [DEC]	–31 d	$x = -31_{10}$.
[x≷y]	80000000 h	$y = 80000000_{16} = 2^{31}$. y ist jetzt ganzzahlig derart, daß $2^{31} \leq \lvert y \rvert < 2^{32}$ und $y \cdot 2^{-31} = 1$ gilt.

Andere Auswirkungen beim Umschalten in den Integer-Modus

Wenn Sie vom dezimalen Gleitkomma-Modus in den Integer-Modus schalten, sollten Sie folgendes beachten:

- Wortlänge *bleibt* 56 Bit.
- Stack und Speicherregister werden *nicht* gelöscht. Die Speicherregisterwerte, die nicht im Integer-Modus abgespeichert wurden, werden verfälscht.*
- Der Komplement-Modus bleibt erhalten.

---

* Sie können diese Werte *in ihrer Originalform wiedererhalten*, wenn Sie den Rechner in den Gleitkomma-Modus zurückschalten.

Gleitkomma-Arithmetik

Funktionen

Alle Funktionen des Integer-Modus ([+], [-], [×], [÷], [√x] arbeiten auch im dezimalen Gleitkomma-Modus. Die Funktionen [÷] und [√x] sind im Gleitkomma-Modus nicht auf ganzzahlige Ergebnisse beschränkt.

Die Funktion [1/x] arbeitet nur im dezimalen Gleitkomma-Modus und berechnet den Reziprokwert der Zahl im X-Register.

Der Out-Of-Range Flag

Im dezimalen Gleitkomma-Modus beeinflussen nur die arithmetischen Operatoren den Zustand von Flag 5 (Out-Of-Range). Bei jeder Ausführung einer arithmetischen Operation wird Flag 5 gesetzt oder gelöscht. Zusätzlich wird Flag 5 auch durch [FLOAT] {0 bis 9, [·]} beeinflusst.

Der Zustand des Carry-Flag (Flag 4) wird im dezimalen Gleitkomma-Modus nicht verändert.

Desaktivierte Funktionen im dezimalen Gleitkomma-Modus

Ganz allgemein sind alle Anzeigen- und Zahlenkontrollfunktionen sowie die Bitmanipulationsfunktionen im dezimalen Gleitkomma-Modus nicht aktiviert. In Anhang B finden Sie eine vollständige Auflistung aller im dezimalen Gleitkomma-Modus desaktivierten Funktionen.

Dezimaltrennzeichen und Zifferntrennzeichen

Der HP-16C benutzt in der Grundeinstellung einen Punkt als Dezimaltrennzeichen und ein Komma als Zifferntrennzeichen zwischen Gruppen zu je drei Ziffernstellen. Mit der Tastenfolge [ON]/[·] können Sie den Rechner auf die in den deutschsprachigen Ländern gängige umgekehrte Notation umstellen.

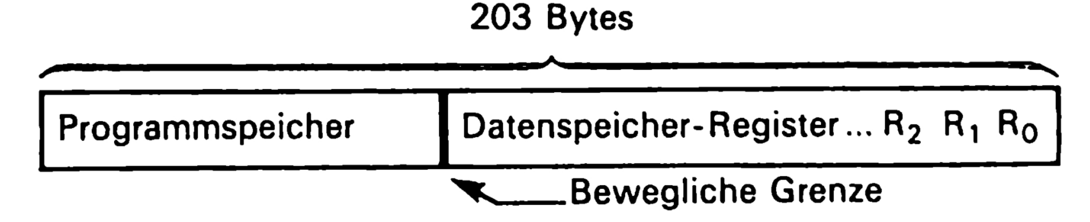
1. Schalten Sie den Rechner aus.
2. Drücken und halten Sie die Taste [ON].
3. Drücken und halten Sie die Taste [·].
4. Lassen Sie zuerst [ON], dann [·] wieder los.

Abschnitt 6

Speicher

Speicheraufteilung

Der Speicher des HP-16C kann in *Register zur Datenspeicherung und in Programmzeilen* zur Programmspeicherung aufgeteilt werden. Insgesamt besteht der Speicher aus 203 Bytes (1 Byte = 8 Bit), die zu Anfang- vollständig der Datenspeicherung zugeordnet sind. Sobald Sie jedoch Programmbefehle eingeben, (siehe Abschnitt 7, Grundlagen der Programmierung), wird dem Programmspeicher *automatisch Speicherplatz zugeteilt.*



Umwandlung von Datenspeicher-Registern in Programmspeicher

Die automatische Zuteilung von Speicherplatz für Programmzeilen erfolgt schrittweise in Segmenten zu 7 Bytes, die jeweils sieben Programmzeilen aufnehmen können. Mit Eingabe der ersten und jeder siebten nachfolgenden Programmzeile werden sieben Bytes Datenspeicher in Programmspeicher umgewandelt.

Automatische Speicheraufteilung (Bytes)

Eingegebene Programmzeilen*	Zugewiesener Programmspeicher	Zugewiesener Datenspeicher
0 bytes	0 Bytes	203 Bytes
1 bis 7	7	196
8 bis 14	14	189
15 bis 21	21	182
⋮	⋮	⋮
190 bis 196	196	7
197 bis 203	203	0
*eine Zeile entspricht einem Byte.		

* Zu Anfang bedeutet: Vor dem ersten Einschalten oder nach einem Löschen des Permanentspeichers.

Die Anzahl der Bytes, die der Daten- bzw. Programmspeicherung zugeteilt sind, ist daher jeweils ein Vielfaches von sieben.

Hinweis: Der Rechner wandelt Datenspeicher-Register in umgekehrter numerischer Reihenfolge in Programmspeicher um (beginnend mit dem Register mit der höchsten Adresse). Alle in einem Datenspeicher-Register *gespeicherte Information geht bei dessen Umwandlung in Programmspeicher verloren.*

Umwandlung von Programmspeicher in Datenspeicher-Register

Einmal eingegebene Programmbefehle sind gegen eine zufällige Löschung geschützt. Die Rückumwandlung von Programm- in Datenspeicher ist nur durch explizite Löschung von Programmbefehlen, entweder einzeln oder auch insgesamt (durch CLEAR [PRGM] oder Löschen des Permanentspeichers), möglich. Gelöschte Bereiche des Programmspeichers werden (in Blöcken zu 7 Bytes) wieder der Datenspeicherung zugeteilt.

Auf diese Weise wird der *unbeabsichtigte* Verlust von Programmbefehlen verhindert. Wenn Sie versuchen, ein Datenspeicher-Register zu adressieren, das in Programmzeilen umgewandelt ist, meldet der Rechner **Error 3** (*nicht vorhandenes Datenspeicher-Register*).

Größe der Datenspeicher-Register

Jedes Register kann ein Wort aufnehmen, die Größe des Registers hängt also von der derzeitig eingestellten Wortlänge ab.– Die Größe eines Datenspeicher-Registers ist *immer das kleinste Vielfache von 4 Bits* (einem Halb-Byte) größer oder gleich der gegebenen Wortlänge. Eine Wortlänge von 13,14, 15 oder 16 Bit wird also die Größe der Datenspeicher-Register auf 16 Bit (2 Bytes) festlegen.

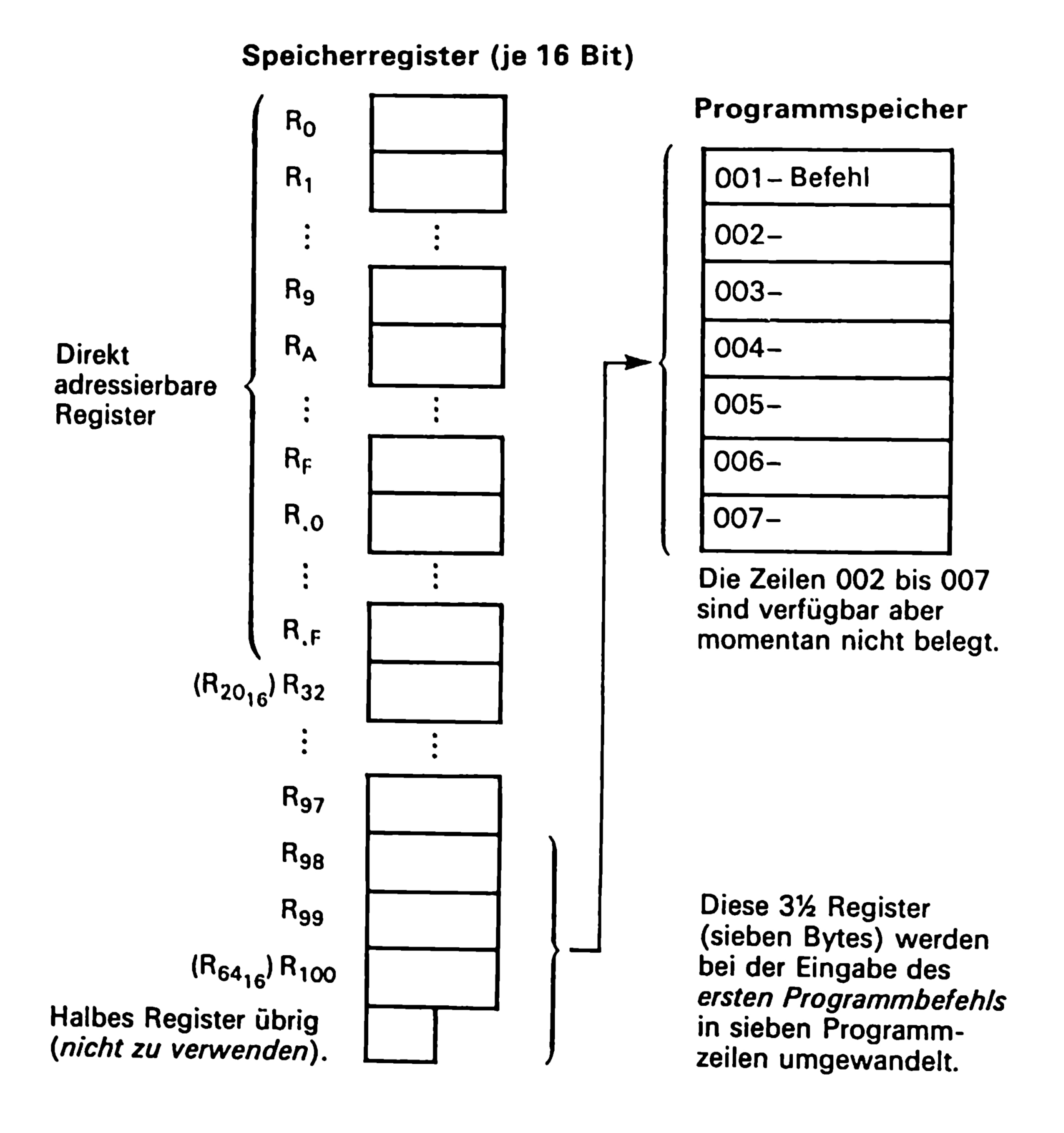
Im dezimalen Gleitkomma-Modus ist die Wortlänge und damit die Größe der Datenspeicher-Register automatisch auf 56 Bit (7 Bytes) eingestellt.

Die mögliche Gesamtzahl von Datenspeicher-Registern ist gleich der Anzahl verfügbarer Bytes (203 Bytes minus Bytes des Programmspeichers), geteilt durch die Anzahl der Bytes pro Register. Ist die derzeitige Wortlänge 16 Bit, dann umfaßt jedes Register 2 Bytes (16 Bits).

* Außer dem Indexregister, das konstant aus 68 Bit besteht und nicht in Programmspeicher umgewandelt werden kann.

Bei gelöschtem Programmspeicher (alle 203 Bytes sind dem Datenspeicher zugeteilt), stehen Ihnen dann 101 Datenspeicher-Register zur Verfügung (203/2 = 101 ½). *Halbe Register sind nicht für Datenspeicherung verfügbar.*

Bei einer Wortlänge von 16 Bit und einer gespeicherten Programmzeile würde die Speicheraufteilung folgendermaßen aussehen:



Bei der Abspeicherung einer achten Programm-Anweisung werden weitere 3½ Register in Programmspeicher umgewandelt.

Damit verbleiben 94½ Register, von denen nur 94 effektiv als Datenspeicher-Register verfügbar sind. (Das verbleibende halbe Halb-Byte wird bei der nächsten Programmspeicher-Umwandlung mitbenutzt.)

Anzeigen der Speicheraufteilung ([MEM])

Sie können durch Drücken von [f] [MEM] die momentane Speicheraufteilung kurzfristig anzeigen. Die Anzeige erscheint solange, wie Sie [MEM] gedrückt halten, und enthält die folgenden Informationen:

P – *B*   r – *RRR*

B ist die Anzahl der im derzeitigen 7-Byte Segment noch verfügbaren Programmzeilen (bevor der Datenspeicherbereich um weitere 7 Bytes verringert wird). $0 \leqslant B \leqslant 6$.

RRR ist die Gesamtanzahl der derzeitig verfügbaren Datenspeicher-Register, $0 \leqslant RRR \leqslant 406$, wobei *RRR* immer dezimal angezeigt wird. (Die kleinstmögliche Registerlänge ist 4 Bit, die maximale Registeranzahl also 406.)

Für die im Diagramm auf der letzten Seite dargestellte Speicheraufteilung würde die Funktion [MEM] die folgende Anzeige liefern:

Tastenfolge	**Anzeige**		
[ON]/[−] [BSP]		0 h	Löscht den Permanentspeicher (und damit den Programmspeicher); Wortlänge ist 16 Bit.
[f] [MEM] (gehalten) (losgelassen)	P-0	r-101 0 h	0 weitere Zeilen, bevor 7 Bytes in Programmspeicher verwandelt werden. 101 Datenregister verfügbar.
[g] [P/R]	000–		Schaltet den Rechner in den Programm-Modus, Anzeige von Zeile 000.
1	001–	1	Speichert eine Programmzeile.

Tastenfolge	Anzeige		
[f] [MEM]	P–6 001–	r–098 1	Sechs weitere Zeilen (Bytes), bevor weitere 7 Bytes in Programmspeicher umgewandelt werden. 98 Datenregister verfügbar.
[f] CLEAR [PRGM]	000–		Programmspeicher gelöscht.
[g] [P/R]		0 h	Verlassen des Programm-Modus.

Speicherregister-Operationen

Beim Abspeichern und Abrufen von Daten wird der Inhalt des X-Registers in ein Datenspeicher-Register kopiert bzw. umgekehrt. Die Anzahl der insgesamt verfügbaren Datenspeicher-Register hängt, wie oben erwähnt, von der derzeitigen Wortlänge und der Größe des noch verfügbaren Speichers ab.

Bis zu 32 Datenspeicher-Register können *direkt* adressiert werden (0 bis 9 und A bis F, .0 bis .9 und .A bis .F), auf die restlichen Register (bis zu 69 bei Wortlänge 16) kann nur *indirekt* mit Hilfe des Indexregisters zugegriffen werden. Da die Register mit der höchsten Adresse zuerst in Programmspeicher umgewandelt werden, ist es sinnoll, Ihre Daten in Registern mit der jeweils niedrigsten Adresse abzulegen (siehe dazu auch das Diagramm auf der Innenseite des Rückumschlags).

Direktes Speichern und Abrufen von Daten

Um Daten in einem direkt adressierbaren Datenspeicherregister *abzuspeichern* oder *abzurufen*, drücken Sie [STO] {0 bis F, .0 bis .F, [I]} oder [RCL] {0 bis F, .0 bis .F, [I]}. Die Operation [STO] ersetzt den Inhalt des adressierten Registers durch eine Kopie des X-Registers; die Operation [RCL] bewirkt einen Stack Lift (falls freigegeben) und kopiert den Inhalt des adressierten Registers in das X-Register.

Wenn Sie versuchen, ein nicht vorhandenes Register (einschließlich eines in Programmspeicher umgewandelten Registers) aufzurufen, zeigt der Rechner die Meldung **Error 3** an. Denken Sie auch daran, daß Sie gespeicherte Daten verlieren können, wenn Datenspeicher-Register *automatisch* in Programmspeicher umgewandelt werden, und daß die Register mit der höchsten Adresse zuerst umgewandelt werden.

Beispiel: Speichern Sie 10^8 in R_0, rufen Sie diesen Wert zurück und multiplizieren Sie ihn mit 2.

Tastenfolge	Anzeige	
[f] [FLOAT] 0		Anzeige hängt von den letzten Inhalten von X und Y ab.
[f] [EEX] 8	1 08	
[STO] 0	100,000,000.	
[BSP]	0.	
[RCL] 0	100,000,000.	
2 [x]	200,000,000.	

Einfluss der Wortlänge auf die Registerinhalte

Wenn Sie die momentane Wortlänge ändern, werden die Inhalte der Stackregister beeinflusst (siehe Abschnitt 3, Wortlänge), *nicht aber die Inhalte der Speicherregister.* Bei einer Änderung der Wortlänge ändern sich jedoch die Größe und die Grenzen (und damit die Anzahl) der Speicherregister. Deshalb wird das Abrufen von Daten in ihrer ursprünglichen Form im allgemeinen nicht möglich sein – *bis Sie die ursprüngliche Wortlänge wiederherstellen.* Sie können also zeitweise mit veränderter Wortlänge rechnen, und beim Zurückkehren zur ursprünglichen Wortlänge stehen Ihre zuvor abgespeicherten Daten wieder unverfälscht zur Verfügnung.

Beispiel: Der folgende Sequenz verdeutlicht, was mit den Daten in R_0 und R_1 bei einer Verdoppelung der Wortlänge (von 16 auf 32 Bit) und anschließender Wiederherstellung des Ausgangszustandes passiert.

Tastenfolge	Anzeige	
[HEX] 10 [f] [WSIZE]		Hexadezimal-Modus.
[f] CLEAR [REG]		Löscht alle Datenspeicher-Register.
1234 [STO] 0	1234 h	Speichert in R_0.
5678 [STO] 1	5678 h	Speichert in R_1.
20 [f] [WSIZE]	5678 h	Verdoppelt die Wortlänge.
[RCL] 0	56781234 h	Momentanes Register R_0 ist eine Verkettung der alten Register R_0 und R_1.
[RCL] 1	0 h	Momentanes Register R_1 ist leer.
10 [f] [WSIZE]	0 h	Setzt wieder die ursprüngliche Wortlänge.
[RCL] 0	1234 h	Ursprüngliche Inhalte von R_0
[RCL] 1	5678 h	und R_1 sind wieder vorhanden.

Beim Umschalten des Rechners vom Integer- in dezimalen Gleitkomma-Modus und umgekehrt werden die Register*inhalte nicht* verändert. Weil die beiden Modi aber verschiedene interne Darstellungen besitzen, werden die *Werte* dieser Inhalte beim Umschalten verfälscht. Sie erhalten Ihre ursprünglichen Werte aber wieder, wenn Sie in Originalmodus und -Wortlänge zurückschalten.

Löschen von Datenspeicher-Registern

Mit [f] CLEAR [REG] (*clear registers*) löschen Sie die Inhalte aller Datenspeicher-Register (Stack und LAST X werden nicht gelöscht). Um ein einzelnes Datenregister zu löschen, speichern Sie den Wert 0 in diesem Register. Beim Löschen des Permanentspeichers werden alle Datenspeicher- und Stackregister mitgelöscht.

Das Indexregister

Das Indexregister R_I ist ein permanentes Speicherregister, das zur *indirekten* Adressierung anderer Datenregister, zur *indirekten* Verzweigung zu Programmlabels und zur Speicherung des Schleifenzählers bei der Steuerung von Programmschleifen benutzt werden kann. (Die beiden letzteren Anwendungen werden in Abschnitt 9 erläutert.) Im Gegensatz zu den anderen Datenregistern besteht das Indexregister, unabhängig von der momentanen Wortlänge, *immer aus 68 Bit und wird nicht in Programmspeicher umgewandelt.*

Abgekürzte Tastenfolgen

Wann immer die Taste [I] oder [(i)] auf eine andere Funktionstaste folgt (wie z.B. [STO], [RCL], [GTO] oder [GSB]), kann die Vorwahltaste [f] vor [I] oder [(i)] weggelassen werden, da die Tastenfolge eindeutig ist. Eine solche Tastenfolge wird als *abgekürzte Tastenfolge* bezeichnet. ([STO] [I], zum Beispiel, ist kürzer als [STO] [f] [I], hat aber dieselbe Wirkung.)

Speichern und Abrufen von Werten im Indexregister

Der Zugriff auf den Inhalt des Indexregisters erfolgt mit Hilfe der Funktion [I]: [STO] [I], [RCL] [I], oder [x≷I]. Eine in R_I abgespeicherte Zahl wird im 68-Bit Format, numerisch gleichwertig mit der Zahl im X-Register, dargestellt. Von einer aus dem Indexregister abgerufenen Zahl werden gegebenenfalls die höherwertigen Stellen* abgeschnitten, um sie an die gegebene Wortlänge im X-Register anzupassen.

* Ist die Wortlänge beim Abrufen des Inhalts von R_I kleiner als beim Abspeichern, ist der abgerufene Wert möglicherweise vom Inhalt von R_I verschieden.

Speichern und Abrufen. Das Kopieren von Zahlen in das und aus dem Indexregister unterscheidet sich nicht von anderen Registern und erfolgt über die Tastenfolgen [STO] [I] bzw. [RCL] [I].

Austausch von X und I. Analog zur Funktion [x≷y] tauscht [x≷I] die Inhalte des X- und des Indexregisters.

Indirektes Speichern und Abrufen

Mit Hilfe der Funktion [(i)] können Sie *indirekt* auf Datenspeicher-Register zugreifen: [STO] [(i)], [RCL] [(i)], oder [x≷(i)]. Der Absolutwert der Zahl in R_I bestimmt die Adresse des Datenspeicher-Registers*. (Im dezimalen Gleitkomma-Modus wird nur der ganzzahlige Anteil benutzt.) Die folgende Tabelle zeigt den Zusammenhang zwischen R_I und den Adressen der Datenspeicher-Register.

Indirekte Adressierung

Inhalt von R_I		Durch [(i)] adressiertes Register
0	(0_{16})	R_0
⋮	⋮	⋮
9	(9_{16})	R_9
10	(A_{16})	R_A
⋮	⋮	⋮
15	(F_{16})	R_F
16	(10_{16})	$R_{.0}$
⋮	⋮	⋮
31	($1F_{16}$)	$R_{.F}$
32	(20_{16})	R_{32} ($= R_{20_{16}}$)
33	(21_{16})	R_{33} ($= R_{21_{16}}$)
⋮	⋮	⋮

Die Funktionen [STO] [(i)], [RCL] [(i)] und [x≷(i)] sind die *einzige* Möglichkeit, auf die hinter den ersten 32 Registern liegenden Datenspeicher-Register zuzugreifen, sie können jedoch *auch* zur Adressierung der ersten 32 Register (0 bis .F) benutzt werden, wie die Tabelle zeigt.

* Bei der Berechnung des Absolutwerts wird der momentane Komplement-Modus und eine Wortlänge von 68 Bit benutzt.

Beispiel: Speichern Sie die Zahl 3 in R_{326}. Um einen Datenregister mit dieser hohen Adresse zur Verfügung zu haben, darf die Wortlänge maximal 4 Bit betragen (203 Bytes / 0,5 Bytes = 406 Register). Zur Abspeicherung einer derart großen Adresse im Indexregister muß die Wortlänge jedoch größer sein. (Das Indexregister wird zur Adressierung aller Register oberhalb 32 (= .F) benötigt). Um den Wert 3 in R_{326} abzuspeichern müssen Sie deshalb die Wortlänge zweimal ändern. Nehmen wir an, der Rechner steht auf Wortlänge 4.

Tastenfolge	**Anzeige**	([STATUS]:2-04-0000)
[DEC] 0 [f] [WSIZE]		Setzt Wortlänge auf 64 Bit.
326	**326 d**	
[STO] [I]	**326 d**	Speichert 326_{10} in R_I.
4 [f] [WSIZE]	**6 d**	Jetzt sind 406 Register zu je 4 Bit verfügbar.
3 [STO] [(i)]	**3 d**	Speichert den Wert 3 in R_{326}.
[BSP]	**0 d**	Löscht die Anzeige.
[RCL] [(i)]	**3 d**	Ruft den Inhalt von R_{326} ab.

Teil II

Programmierung des HP-16C

Abschnitt 7

Grundlagen der Programmierung

Die nächsten drei Abschnitte sind der Programmierung des HP-16C gewidmet. In diesen Abschnitten werden zunächst grundlegende Techniken (das «Handwerkszeug») besprochen, dann Anwendungsbeispiele für diese Techniken gegeben, und zum Schluß Details und Ergänzungen angefügt (Weitere Informationen). Damit brauchen Sie nur soweit zu lesen, bis Sie sich mit der Programmierung Ihres HP-16C vertraut fühlen.

Das Handwerkszeug

Erstellen eines Programms

Die Programmierung des HP-16C ist sehr einfach; Sie brauchen nur dieselbe Tastenfolge abzuspeichern, die das Problem manuell lösen würde («Tastenprogrammierung»). Um aus einer Serie von Rechenschritten ein Programm zu machen, sind zwei weitere Schritte erforderlich: die Entscheidung, wo und in welcher Form Daten einzugeben sind, und das Laden und Speichern des Programmes. Zusätzlich besteht die Möglichkeit, das Programm Entscheidungen und Iterationen mit Hilfe von bedingten und unbedingten Verzweigungen ausführen zu lassen.

Wir werden im Verlauf der Beschreibung der Programmier-Grundlagen ein Programm zusammenstellen, das zwei 16-Bit Worte in den Registern X und Y zu einem 32-Bit Wort im X-Register verkettet.

Laden eines Programmes

Programm-Modus. Drücken Sie [g] [P/R] (*program/run*) um den Rechner in Programm-Modus zu schalten (Statusanzeige **PRGM** erscheint). Die meisten Funktionen werden nicht ausgeführt, sondern gespeichert, wenn sie im Programm-Modus eingetastet werden.

Tastenfolge	Anzeige	
[g] [P/R]	000–	Schaltet den Rechner in Programm-Modus. In der Anzeige erscheint die Zeilennummer 000 und die Statusanzeige **PRGM**.

Tastenfolgen werden im Programmodus zu Programmanweisungen, die Programmzeilen belegen. Diese numerierten Zeilen zeigen die Position des Rechners im Programmspeicher an. Zeile 000 steht am Anfang des Programmspeichers und kann nicht zur Speicherung eines Befehls benutzt werden; die erste Anweisung steht daher in Zeile 001. Eine Programmzeile (außer 000) wird erst erzeugt, wenn eine Anweisung dort abgespeichert wird.

Programme beginnen gewöhnlich mit Zeile 001; Sie können ein Programm aber grundsätzlich mit jeder beliebigen existierenden Zeile beginnen. Bei der Eingabe neuer Programmbefehle bleiben bereits vorhandene Programmbefehle erhalten und werden unter Erhöhung ihrer Zeilennummern im Programmspeicher nach unten verschoben.

Der Programmanfang. Das Löschen des Programmspeichers löscht alle Programme im Speicher und setzt den Rechner auf Zeile 000. Drücken Sie dazu [f] CLEAR [PRGM] *im Programm-Modus.*

Sie können den Rechner auf Zeile 000 positionieren, ohne dabei den Programmspeicher zu löschen, indem Sie im *Run-Modus* [f] CLEAR [PRGM] oder [GTO] [·] 000, oder im *Programm-Modus* [GTO] [·] 000 eintasten. (Die Anweisung [GTO] [·] kann nicht gespeichert werden.)

Eine [LBL] (*Label*) Anweisung, d.h. [g] [LBL] gefolgt von einem numerischen oder alphanumerischen Label {0 bis 9, A bis F}, wird zur Markierung des Anfangs eines Programms oder Unterprogramms verwendet. Die Benutzung von Labels ermöglicht Ihnen, schnell ein bestimmtes Programm anzuwählen und zu starten.

Tastenfolge	**Anzeige**	
[f] CLEAR [PRGM]	**000–**	Löscht den Programmspeicher und setzt den Rechner auf Zeile 000.
[g] [LBL] A	**001–43,22, A**	Tastencode für Label «A».

Speichern eines Programms. Ein Programm besteht aus derselben Tastenfolge, die Sie auch zur manuellen Lösung eines Problems anwenden würden. Im Programm-Modus gedrückte Tasten werden als Programmanweisungen gespeichert.* In der Anzeige erscheinen danach eine Zeilennummer und ein oder mehrere *Tastencodes.* Tastencodes sind ein- oder zweistellige Zahlen, die die Position der Tasten auf dem Tastenfeld angeben (siehe Seite 77).

* Mit Ausnahme der *nichtprogrammierbaren Funktionen,* die auf Seite 81 zusammengestellt sind.

Beispiel: Nachfolgend ist das bereits erwähnte Verkettungsprogramm aufgelistet. Die Programmzeilen 002 bis 008 verketten die beiden 16-Bit Zahlen in den Registern X und Y zu einer 32-Bit Zahl. Das ursprünglich im X-Register stehende Wort liefert die höherwertigen Bits des Resultats.

Tastenfolge	Anzeige		
[HEX]	002–	23	
2	003–	2	Verdoppelt die Wortlänge von 16 auf 32; links der Zahlen in X- und Y-Register sind 16 zusätzliche Bits verfügbar.
0	004–	0	
[f] [WSIZE]	005–	42 44	
[g] [LSTx]	006–	43 46	Lädt die alte Wortlänge aus LAST X nach.
[f] [SR]	007–	42 b	Halbiert die Wortlänge (16).
[f] [RLn]	008–	42 E	Verschiebt die Zahl um 16 Bit nach links.
[f] [OR]	009–	42 40	Verkettet die Inhalte von X und Y durch logische ODER Operation.

Beenden eines Programms

- Die Anweisung [g] [RTN] (*return*) beendet ein Programm, setzt den Rechner auf Zeile 000 und stoppt*. Diese Anweisung ist nach dem letzten Programm im Speicher nicht erforderlich, da das Ende des Programmspeichers automatisch einen [RTN] Befehl enthält.
- Die Anweisung [R/S] (*run/stop*) hält ein Programm an, *ohne* zu Zeile 000 zurückzukehren.

Tastenfolge	Anzeige		
[g] [RTN]	010–	43 21	Nicht notwendig beim letzten Programm im Speicher.

* Sofern keine anstehenden Unterprogrammrücksprünge mehr vorhanden sind; siehe Seite 94.

Ausführen eines Programms

Run-Modus. Nach der Eingabe des Programms schalten Sie den Rechner mit [g] [P/R] in den Run-Modus zurück. Sie können Programme nur im Run-Modus ausführen.

Tastenfolge	Anzeige	
[g] [P/R]		Run-Modus; Statusanzeige **PRGM** erlischt. Die Anzeige enthält das letzte Ergebnis.

Die Position des Rechners im Programmspeicher wird beim Übergang zwischen Run- und Programm-Modus nicht verändert. Nach dem Ausschalten «erwacht» der Rechner immer im Programm-Modus.

Starten des Programms. Drücken Sie im Run-Modus [GSB] *Label.* Diese Tastenfolge adressiert das spezifizierte Programm und startet die Ausführung. In der Anzeige blinkt die Meldung **«running»**. (Die Taste [GSB] wird in anderem Zusammenhang auch zum Aufruf von Unterprogrammen benutzt.)

Tastenfolge	Anzeige	([STATUS]:2-16-0000)
[HEX] FFFE [ENTER]	**FFFE h**	Lädt die erste Zahl in Register X und Y.
DDDC	**dddC h**	Lädt die 2. Zahl ins X-Register. Diese Ziffern werden zur rechten Hälfte der neuen Zahl werden.
[GSB] A	**dddCFFFE h**	Die verkettete Hexadezimalzahl.
[f] [STATUS]	**2–32–0000**	Wortlänge ist nun 32 Bit.

Alternativ können Sie den Rechner auch mit [GTO] [·] *nnn* (dreistellige Zeilennummer) oder [GTO] *Label* auf die gewünschte Zeile positionieren und dann die Ausführung mit [R/S] starten.

Vorübergehende Unterbrechung der Programmausführung

Mit Hilfe des Programmbefehls [f] [PSE] (*Pause*) können Sie den Programmablauf *kurzzeitig* anhalten, um ein Zwischenergebnis anzuzeigen. Für längere Pausen benutzen Sie einfach diese Funktion mehrmals hintereinander.

Die Anweisung [R/S] unterbricht die Programmausführung für unbestimmte Zeit mit der auf [R/S] folgenden Programmzeile. Sie können die Programmausführung von dort aus wieder fortsetzen, wenn Sie im Run-Modus [R/S] drücken.

Dateneingabe

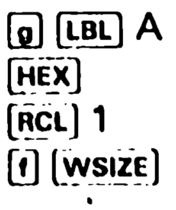
Jedes Programm muß berücksichtigen, wann und in welcher Form Daten zur Verfügung gestellt werden. Die Dateneingabe kann vor der Programmausführung oder während geplanter Programmunterbrechungen erfolgen.

Vorhergehende Eingabe.

1. Sie können die Daten (mit [STO]) in einem Datenspeicher-Register abspeichern, von wo sie (mit einem [RCL] im Programm) während der Programmausführung wieder abgerufen werden.

   Das Verkettungsprogramm könnte beispielsweise die Wortlänge von einem Register *abrufen*, anstatt diesen Wert im Programm selbst zu enthalten:

   Tastenfolge



Ruft Wortlänge aus R_I ab.

2. Werden Daten in den ersten Programmzeilen benötigt, können Sie sie vor dem Programmstart in den Stack speichern. Das Programm sollte nicht mit dem Befehl [ENTER] beginnen – die Funktionen [LBL], [GSB] und [GTO] beenden die Zahleneingabe und geben den Stack frei.* Das letzte Programmbeispiel illustriert diese Technik.

   Der Stack ermöglicht Ihnen, mehr als eine Variable vor dem Programmstart vorzugeben. Wenn Sie die Stackbewegungen bei nachfolgenden Operationen berücksichtigen, können Sie in alle vier Register Variablen zur Benutzung innerhalb des Programms laden.

   Diese Technik wurde auf Seite 75 benutzt, wo die beiden benötigten Daten vor dem Programmstart in das X- und Y-Register geladen wurden.

* Beachten Sie jedoch, daß [R/S] den Stack nicht freigibt. In Anhang B finden Sie eine vollständige Auflistung der Funktionen, die den Stack nicht freigeben oder sperren.

Eine Variante könnte darin bestehen, die beiden Zahlen in den Registern Y und Z und die Wortlänge im X-Register abspeichern. Mit diesen Werten im Stack könnte der Programmanfang wie folgt aussehen:

Tastenfolge

g LBL A	Wortlänge im X-Register.
HEX	
f WSIZE	Der Stack fällt. Das höherwertige Wort (aus Y) ist nun im X-Register; das niederwertige Wort (aus Z) ist im Y-Register.
⋮	

Direkte Eingabe. Bei dieser Technik werden die Daten während der Programmausführung erst in dem Moment eingegeben, wenn sie benötigt werden. Setzen Sie eine R/S (*run/stop*) Anweisung an die Stellen im Programm, wo Daten benötigt werden, so daß die Programmausführung dort angehalten wird. Geben Sie die Daten ein, und drücken Sie danach R/S, um die Programmausführung wieder fortzusetzen.

Programmspeicher

Der HP-16C verfügt über 203 Bytes für Programm- und Datenspeicher. Bei Bedarf wird Datenspeicher automatisch in Programmspeicher umgewandelt, jeweils in Blöcken zu sieben Bytes (siehe dazu Abschnitt 6).

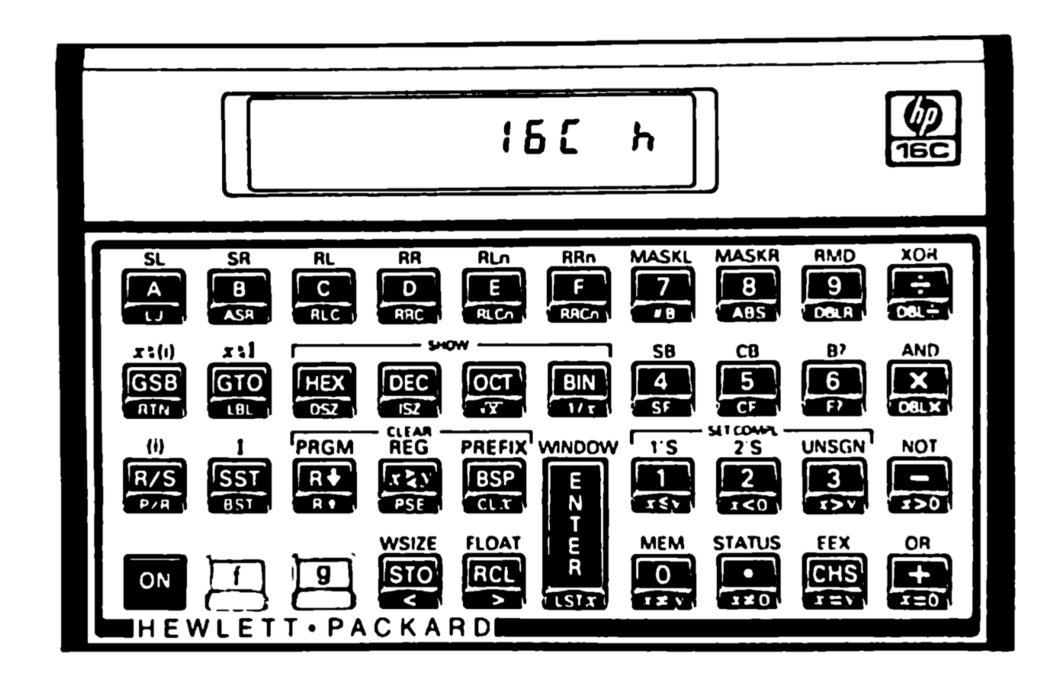
Programmbefehle und Tastencodes

Alle Ziffern- und Funktionstasten sowie das Dezimaltrennzeichen werden als Befehle betrachtet und belegen eine Zeile (ein Byte) im Programmspeicher. Befehle, die Vorwahltasten beinhalten (wie f, STO, GTO und LBL), benötigen ebenfalls nur eine Zeile.

Jeder Taste auf dem Tastenfeld des HP-16C wird im Programm-Modus ein ein- oder zweistelliger Tastencode zugeordnet.

Die erste Ziffer eines zweistelligen Tastencodes bezieht sich auf die Reihen (1 bis 4 von oben nach unten), die zweite Ziffer auf die Spaltenposition (1 bis 9 von links nach rechts) der Taste auf dem Tastenfeld. Der Tastencode einer Zifferntaste (einschließlich A bis F) besteht aus dieser Ziffer selbst. Zum Beispiel:

Befehl	Code	
g LBL 1	001–43,22, 1	
f SET COMPL UNSGN	002– 42 3	SET COMPL UNSGN hat den Code «3»



42: Reihe vier, Spalte zwei

Beispiel

Das folgende Programm benutzt die doppeltgenauen Funktionen (siehe Seite 52 bis 55), um große Zahlen aus *beliebigen* Zahlensystemen zu multiplizieren und um ein auf bis zu 39 Stellen genaues *dezimales* Ergebnis (kleiner als $10^{19} \times 2^{64}$) zu berechnen. Die höherwertigen Stellen des doppeltgenauen Resultats werden im X-Register, die niederwertigen im Y-Register abgelegt.

Da die doppeltgenauen Funktionen intern *binär* arbeiten, ist es notwendig, zusätzliche Vorbereitungen zu treffen (Division durch 10^{19}, dem größtmöglichen in einem Register darstellbaren Exponenten zur Basis 10), um ein sinnvolles *dezimales* Ergebnis zu erhalten.

Tastenfolge	Anzeige	
f CLEAR PRGM		Setzt den Rechner auf Zeile 000 *ohne* den Programmspeicher dabei zu löschen. (Löschung erfolgt nur im Programm-Modus.)
g P/R	000–	Programm-Modus. Statusanzeige **PRGM** erscheint.
g LBL 1	001–43,22, 1	

Tastenfolge	Anzeige		
[f] SET COMPL [UNSGN]	002–	42 3	Größerer positiver Wertebereich durch Wegfall des Vorzeichenbits.
[g] [DBLx]	003–	43 20	Doppeltgenaue Multiplikation der Inhalte der Register X und Y.
[STO] 1	004–	44 1	Speichert die höherwertigen Stellen des Ergebnisses in R_1.
[x⇄y]	005–	34	
[STO] 2	006–	44 2	Speichert die niederwertigen Stellen des Ergebnisses in R_2.
[x⇄y]	007–	34	
[RCL] 0	008–	45 0	Abruf des Divisors (hier der größtmöglichen Potenz von 10).
[g] [DBLR]	009–	43 9	
[RCL] 2	010–	45 2	Niederwertige Stellen des Produkts.
[RCL] 1	011–	45 1	Höherwertige Stellen.
[RCL] 0	012–	45 0	Divisor.
[g] [DBL÷]	013–	43 10	
[DEC]	014–	24	Sichert dezimale Anzeige des Resultats.
[g] [RTN]	015–	43 21	

Um das Programm zu starten, setzen Sie den Rechner auf Wortlänge 64 und speichern Sie 10^{19} (die größte im Unsigned-Modus mögliche Potenz von 10) in R_0. Laden Sie anschließend die Zahl 12345678987654 ins X-Register und die Zahl 987654321234567 ins Y-Register.

Tastenfolge	Anzeige	
g P/R		Schaltet den Rechner in den Run-Modus zurück. Anzeige des letzten Ergebnisses.
0 f WSIZE DEC		Setzt maximale Wortlänge (64)
f SET COMPL UNSGN		
10000000 00000000	00000000 .d	
0000 STO 0	00000000 .d	Speichert 10^{19} in R_0.
12345678987654		
ENTER	78987654 .d	Eingabe der zu multiplizierenden Werte.
987654321234567	21234567 .d	Startet das Programm mit dem Label «1»; das resultierende Produkt befindet sich danach in den Registern X und Y. Das höherwertige Wort hat den Wert 1,219,326,320.
GSB 1	19326320 .d	
f WINDOW 1	12 d.	
x≷y	31035818 .d	Das niederwertige Wort hat den Wert 731,267,636,031,035,818. Das vollständige Ergebnis lautet: 1,219,326,320,731, 267,636,031,035,818$_{10}$.
f WINDOW 1	12676360 .d	
f WINDOW 2	73 d.	

Um das Programm mit verschiedenen Faktoren zu wiederholen, geben Sie diese Werte einfach ins X-und Y-Register ein und drücken Sie GSB 1. (Während des Programmablaufs wird Flag 4 gesetzt, weil die Operation DBL÷ einen von Null verschiedenen Rest erzeugt. Dies hat jedoch keine Bedeutung, da das Programm den Rest in Zeile 009 berechnet.

Weitere Informationen

Labels

In einem Programm (oder Unterprogramm) sind Labels Markierungen, die dem Rechner mitteilen, wo die Programmausführung beginnen soll. Es stehen insgesamt 16 Labels (die Zahlen 0 bis 9 und A bis F) zur Kennzeichnung von Programmen und Unterprogrammen zur Verfügung.

Auf einen Suchbefehl (wie [GSB] *Label*) durchsucht der Rechner den Programmspeicher nach dem entsprechenden Label. Dabei beginnt er mit der Programmzeile, in der er sich gerade befindet. Wird das gewünschte Label bis zum Ende des Programmspeichers nicht gefunden, wird die Suche mit Zeile 001 fortgesetzt. Sobald das Label gefunden wird, bricht der Suchlauf ab und die Programmausführung beginnt.

Da der Rechner den Programmspeicher nur in einer Richtung durchsucht, ist es möglich, wenn auch nicht ratsam, ein Label mehrmals zu verwenden. Die Programmausführung beginnt mit der Zeile, in der das spezifizierte Label zuerst gefunden wird.

Abbruch der Programmausführung

Drücken einer beliebigen Taste. Durch Drücken einer beliebigen Taste bricht die Programmausführung sofort nach Beendigung der gerade ausgeführten Operation ab.

Programmabbruch durch Fehler. Wenn der Rechner versucht, eine unzulässige Operation auszuführen, wird die Programmausführung sofort abgebrochen und eine Fehlermeldung angezeigt.

Um Zeilennummer und Tastencode der den Fehler verursachenden Anweisung festzustellen (die Zeile, in der die Ausführung abgebrochen wurde), drücken Sie eine beliebige Taste zum Löschen der Fehler-Anzeige, und schalten Sie den Rechner dann in den Programm-Modus.

Das Auftreten einer Out-Of-Range Bedingung (einschließlich eines Overflows) während der Ausführung eines Programms hält den Programmlauf *nicht* an. Stattdessen werden Flag 5 und die Statusanzeige **G** gesetzt.

Nichtprogrammierbare Funktionen

Im Programm-Modus können fast alle Funktionen des Tastenfeldes als Programmanweisungen gespeichert werden. Dies gilt *nicht* für folgende Funktionen:

[f] CLEAR [PREFIX]	[g] [P/R]	[SST]
[f] CLEAR [PRGM]	[f] [MEM]	[g] [BST]
[ON] / [−]	[f] [STATUS]	[BSP]
[ON] / [·]		[GTO] [·] *nnn*

Abschnitt 8

Programmkorrektur

Der HP-16C ist mit verschiedenen Korrektureinrichtungen ausgestattet, die Ihnen das Ändern gespeicherter Anweisungen oder Programme erleichtern.

Das Handwerkszeug

Programmkorrekturen jeder Art erfordern zwei Schritte: Auffinden der zu korrigierenden Programmzeile und Ausführung der Korrektur (d.h. Einfügungen/Löschungen).

Auffinden einer Zeile im Programmspeicher

Der Befehl GoTo ([GTO]). Die Tastenfolge [GTO] [·] *nnn* setzt den Rechner auf Zeile *nnn*, unabhängig davon, ob der Programm- oder der Run-Modus geschaltet ist. Diese Tastenfolge ist *nicht* programmierbar und dient zum *manuellen* Auffinden einer gegebenen Position im Programmspeicher. Die Zahl *nnn* muß eine dreistellige Zahl zu einer existierenden Programmzeile im Bereich 000 ≤ *nnn* ≤ 203 sein.

Die Anweisung [SST] (*Single Step*). Um den Programmspeicher zeilenweise zu durchlaufen, drücken Sie [SST]. Diese (nichtprogrammierbare) Funktion kann ein Programm beim Durchlaufen ausführen oder nur auflisten.

Im Programm-Modus wird mit [SST] die Speicherposition um eine Zeile erhöht, die dort stehende Anweisung wird angezeigt (aber *nicht* ausgeführt). Wenn Sie die Taste gedrückt halten, läuft der Rechner kontinuierlich durch die Zeilen des Programmspeichers.

Im Run-Modus zeigt [SST] die momentane Programmzeile an, solange die Taste gedrückt bleibt. Beim Loslassen wird die dort stehende Anweisung ausgeführt, das Ergebnis wird angezeigt, und der Rechner springt zu der als nächste auszuführenden Programmzeile. Diese Funktion ist sehr nützlich, um die Abarbeitung eines Programms schrittweise zu verfolgen.

Die Anweisung [BST] (*Back Step*). Um den Rechner im Programmspeicher um eine Zeile zurückzupositionieren, drücken Sie [g] [BST] im Programm- oder Run-Modus. Diese Funktion ist nicht programmierbar. Im Programm-Modus wird der Programmspeicher kontinuierlich rückwärts durchlaufen, solange die Taste [BST] gedrückt bleibt. Die Programmbefehle werden bei dieser Operation *nicht* ausgeführt.

Löschen von Programmzeilen

Programmanweisungen können im *Programm-Modus* mit [BSP] (*back space*) gelöscht werden. Bringen Sie die zu löschende Zeile zur Anzeige und drücken Sie [BSP]. Alle nachfolgenden Programmzeilen werden neu durchnumeriert, damit keine Lücken in der Zeilennummerierung entstehen.

Im Run-Modus beeinflusst [BSP] den Programmspeicher nicht und dient nur zum Löschen der Anzeige.

Einfügen von Programmzeilen

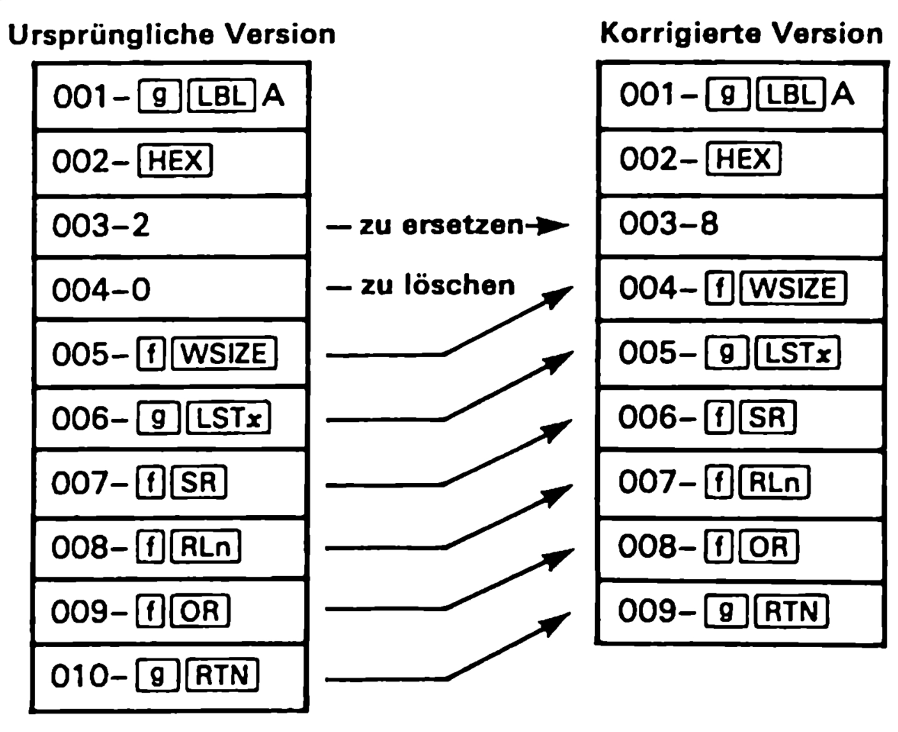
Wenn Sie in ein Programm Anweisungen einfügen wollen, setzen Sie den Programmspeicher auf die Zeile, die der Einfügestelle *voransteht*. Um eine Anweisung zu ändern oder zu ersetzen, löschen Sie diese zuerst, und fügen Sie dann die neue Version ein.

Wenn der Speicher vollständig belegt ist, sind keine Einfügungen möglich; der Rechner meldet in diesem Fall **Error 4**.

Beispiel

Wir werden die Korrektureinrichtungen des HP-16C anhand des Verkettungsprogramms (Seite 74) veranschaulichen. Zuerst wird die Wortlänge von 32 (20_{16}) auf 8 (8_{16}) geändert, dann wird das Programm in Einzelschritten durchlaufen, um die Ausführung zeilenweise zu verfolgen.

Der Korrekturprozess wird in der folgenden Illustration dargestellt. Es wird angenommen, daß das Programm die Zeilen 001 bis 010 des Programmspeichers belegt.



Das Einfügen oder Löschen einer Programmzeile der nachfolgenden Programmzeilen. Wenn Sie das Programm vom Ende her rückwärts korrigieren, bleit die Zeilennumerierung der noch nicht korrigierten Teile des Programms erhalten. Bei den folgenden Korrekturschritten wird unterstellt, daß das Verkettungsprogramm die Zeilen 001 bis 010 des Programmspeichers belegt.

Tastenfolge	**Anzeige**		
[g] [P/R]			Programm-Modus. Die angezeigte Zeilennummer hängt von der letzten Position des Rechners im Programmspeicher ab.
[GTO] [·] (oder [SST])	**004–**	**0**	Positioniert den Rechner auf Zeile 4. (Anweisung hat den Code 0.)
[BSP] [BSP]	**002–**	**23**	Löscht die Zeilen 004 und 003.
8	**003–**	**8**	Fügt eine neue Zeile 003 ein: 8. Die letzten drei Schritte ändern den alten Wert 20 in 8 ab.

Um den Ablauf des revidierten Programmes zu verfolgen, kehren Sie in den Run-Modus zurück, setzen Wortlänge 4, und geben den Wert 7_{16} ins X- und den Wert 6_{16} ins Y-Register ein. Führen Sie danach das Programm mit [SST] Schritt für Schritt aus. Das verkettete Ergebnis mit doppelter Wortlänge sollte 67_{16} sein.

Tastenfolge	**Anzeige**	([STATUS]:2-04-0000)
[g] [P/R]		Run-Modus.
[HEX]		
7 [ENTER]	**7 h**	Niederwertiges 4-Bit Wort.
6	**6 h**	Höherwertiges 4-Bit Wort.
[GTO] [A]	**6 h**	Positioniert den Rechner auf Label A.
[SST] (gehalten)	**001-43,22, A**	Programmzeile 001: Label A.
(losgelassen)	**6 h**	Inhalt des X-Registers.

Erfordert das auszuführende Programm einen anderen Rechnerstatus als den bestehenden, so werden Sie in diesen Fällen fehlerhafte Ergebnisse erhalten. Es ist deshalb ratsam, den Rechner vor der Programmausführung bzw. innerhalb des Programmes zu initialisieren (Löschen von Registern, Setzen der gewünschten Modi usw.). Beachten Sie, daß von einem Programm gesetzte Bedingungen auch alle nachfolgenden Programme beeinflussen.

Zur Initialisierung des Rechners können die folgenden Funktionen benutzt werden: «CLEAR»-Funktionen, die «SET COMPL»-Funktionen, die Zahlenbasis-Modi, Gleitkomma-Modus, [WSIZE], [SF] und [CF].

Abschnitt 9

Programmverzweigungen und Programmsteuerung

Die Verzweigungsoperationen des HP-16C können in einfache (unbedingte) Programmverzweigungen, bedingte Verzweigungen (die Ausführung der Verzweigung hängt von einer bestimmten Bedingung ab) und Unterprogramme eingeteilt werden. Dabei ist das Indexregister, das einen Schleifenkontrollwert aufnehmen kann, für die Verzweigungs- und Schleifensteuerung von großer Bedeutung.

Das Handwerkszeug

Verzweigungen

Die Anweisung Go To ([GTO]). Einfache, d.h. *unbedingte* Verzweigungen, werden mit der Anweisung [GTO] *Label* ausgeführt. In einem laufenden Programm wird nach [GTO] {0 bis 9, A bis F, [I]} die Ausführung mit dem angegebenen Programm (oder Unterprogramm) fortgesetzt.* (Beachten Sie, daß Sie nicht zu einer Zeilen*nummer* verzweigen können.) Der Rechner sucht nach dem gegebenen Label, indem er Programmspeicher von seiner momentanen Position aus nach unten durchläuft und gegebenenfalls die Suche bei Zeile 001 fortsetzt.

[GTO] *Label* kann auch im Run-Modus benutzt werden (d.h., vom Tastenfeld aus), um ein Label im Programmspeicher zu suchen. Das Programm wird in diesem Fall nicht gestartet.

Go To Subroutine ([GSB]). Nach der Anweisung [GSB] *Label* verzweigt der Rechner zu einem mit dem spezifizierten Label gekennzeichneten Unterprogramm. Die Ausführung der nächsten [RTN] Anweisung bewirkt die Rückgabe der Kontrolle an das Hauptprogramm.** Die Ausführung von Unterprogrammen wird später innerhalb dieses Abschnitts erläutert (Seite 94).

* [GTO] [I] und [GSB] [I] sind *abgekürzte Tastenfolgen* (die Vorwahltaste [f] ist nicht erforderlich, siehe Seite 68).

** Falls keine anstehenden Unterprogrammrücksprünge vorhanden sind, stoppt [RTN] die Programmausführung, und der Rechner kehrt zu Zeile 000 zurück.

Indirekte Verzweigungen mit Hilfe des Indexregisters

Sie können mit der Anweisung [GTO] [I] *indirekt* verzweigen, und mit [GSB] [I] *indirekt* ein Unterprogramm aufrufen, wenn Sie zuvor einen *Indexwert* in das Indexregister R_I geladen haben.

Diese Funktionen übergeben die Programmausführung an das Label, das dem *Absolutwert der Zahl im Indexregister* zugeordnet ist (siehe nachfolgende Tabelle).* (Im Gleitkomma-Modus wird nur der ganzzahlige Anteil der Zahl in R_I berücksichtigt.) Sie haben 16 Labels zur Verfügung: 0 bis 9 und A bis F.

Enthält das Index-Register zum Beispiel den Wert -14_{10} ([STATUS]:**2–08–0000**), dann wird die Programmausführung nach einer [GTO] [I] Anweisung bei [LBL] E fortgesetzt ($|-14_{10}| = E_{16}$).

Indirekte Verzweigungen

Inhalt von R_I		Übergabe der Ausführung durch [GTO] [I] oder [GSB] [I] an:
0	(0_{16})	[LBL] 0
⋮	⋮	⋮
9	(9_{16})	[LBL] 9
10	(A_{16})	[LBL] A
⋮	⋮	⋮
15	(F_{16})	[LBL] F

Vergleichsabfragen

Eine weitere Möglichkeit, den Programmablauf zu steuern, besteht in *Vergleichsoperationen* (Ja/Nein-Abfragen), bei denen die Zahl im X-Register entweder mit Null oder mit der Zahl im Y-Register verglichen wird. (Beachten Sie, daß im Einerkomplement-Modus gilt: $-0 = 0$.)

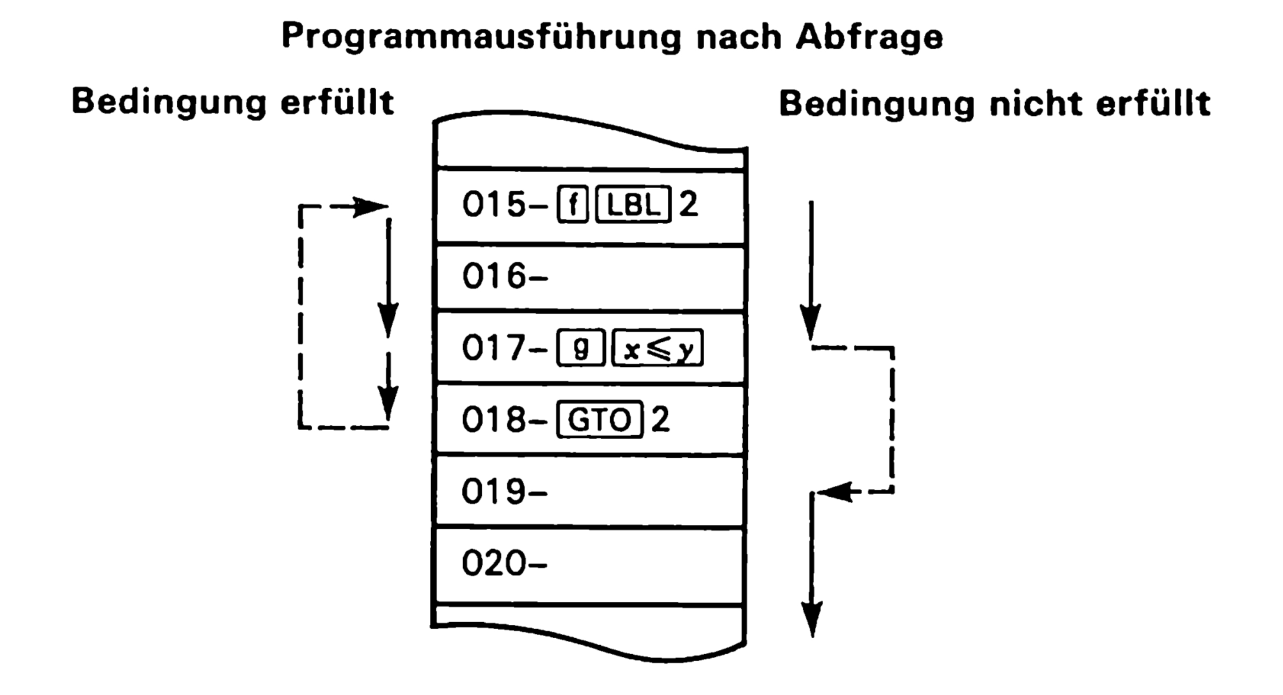
Der HP-16C verfügt über 8 Vergleichsabfragen (die sämtlich mit [g] vorgewählt werden):

[x≤y] [x<0] [x>y] [x>0] [x≠y] [x≠0] [x=y] [x=0]

Nach einem Vergleich folgt das Programm der Regel «*wenn ja, dann*»: wenn die Bedingung erfüllt ist, wird das Programm mit der auf die Abfrage folgenden Anweisung fortgesetzt; andernfalls wird diese übersprungen.

* Der Absolutwert wird mit Wortlänge 68 und im gegebenen Komplement-Modus berechnet.

Es ist sinnvoll, direkt nach der Vergleichsabfrage eine [GTO] Anweisung einzufügen; der Vergleich wird damit zu einer *bedingten Verzweigung*, d.h. [GTO] Anweisung wird nur ausgeführt, wenn die Vergleichsbedingung erfüllt ist.



Abfrage auf gesetzte Flags und Bits

Zusätzliche Abfragen für bedingte Verzweigungen werden durch die Funktionen [F?] (*flag set?*) und [B?] (*bit set?*) ermöglicht. Nach diesen Anweisungen – wie bei den Vergleichen – folgt der Programmablauf der «wenn ja, dann» Regel: bei gesetztem Flag oder Bit wird die darauffolgende Programmanweisung ausgeführt, bei gelöschtem Flag oder Bit übersprungen.

Die einzelnen Flags haben die folgende Bedeutung (siehe Abschnitt 3):

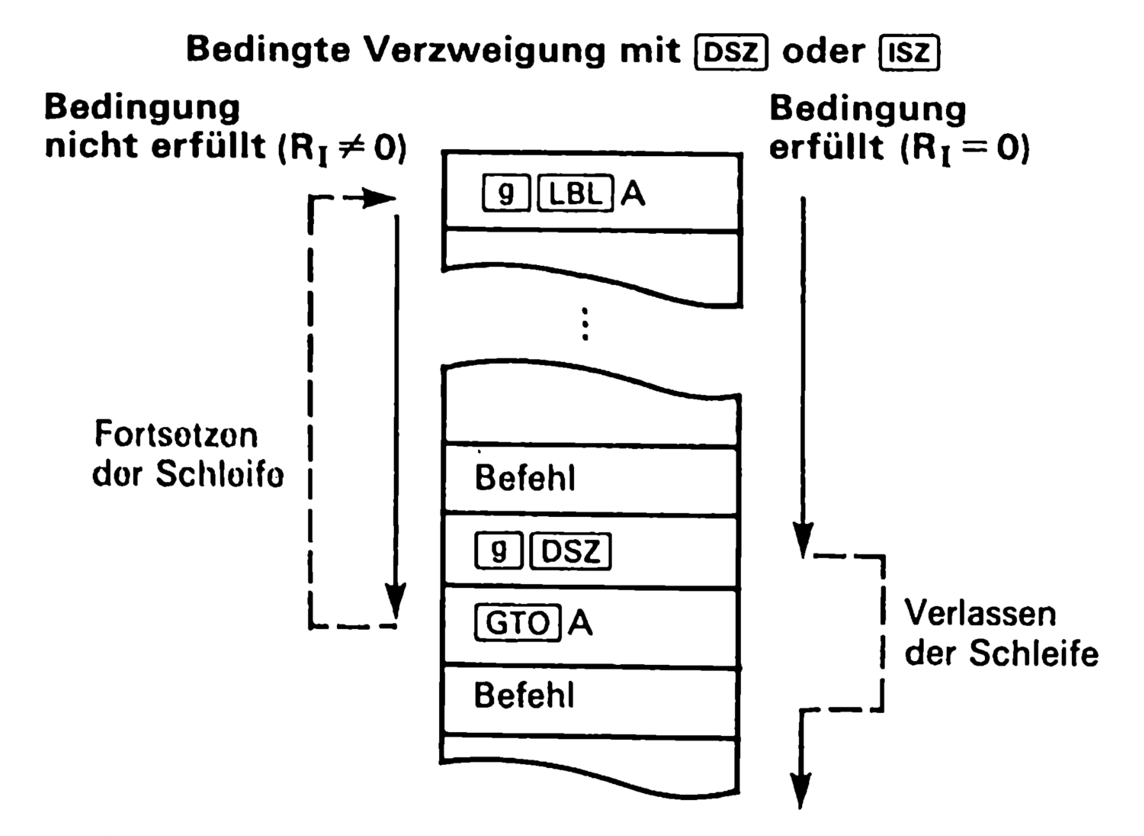
0, 1, 2 } Benutzer-Flags (zur Programmsteuerung).
3 Kontrolliert die Anzeige führender Nullen.
4 Carry oder Borrow Bedingung.
5 Out-Of-Range Bedingung.

Obwohl die Flags 4 und 5 vom Rechner automatisch gesetzt werden, können sie vom Benutzer ebenfalls gesetzt werden. Wenn Sie Flag 4 setzen, erhält das Carrybit den Wert 1. Das Setzen von Flags wird in Abschnitt 3 (Seite 36) und das Setzen von Bits in Abschnitt 4 (Seite 50) beschrieben.

Schleifensteuerung

Die Anweisungen [DSZ] (*decrement and skip if zero*) und [ISZ] (*increment and skip if zero*) dienen der Schleifensteuerung. Beide Anweisungen prüfen und inkrementieren/dekrementieren den Schleifenkontrollwert im Indexregister und überspringen die nächste Programmzeile, wenn der Schleifenkontrollwert den Wert 0 annimmt.

Bei jeder Ausführung dieser Funktionen innerhalb eines Programms wird der Schleifenkontrollwert im Indexregister entweder um den Wert 1 *dekrementiert* ([DSZ]) oder um den Wert 1 *inkrementiert* ([ISZ]). Nimmt der Schleifenkontrollwert den Wert 0 an, wird die darauf folgende Anweisung übersprungen. Dies ermöglicht ein Verlassen der Programmschleife, wenn die übersprungene Zeile eine Verzweigung an den Schleifenanfang enthält.

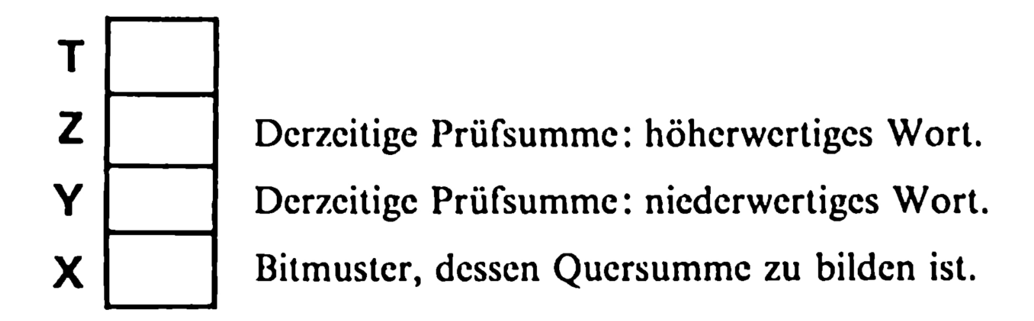


Der Wert in R_I wird im derzeitigen Komplement-Modus interpretiert und positiv oder negativ, im Integer- oder Gleitkommaformat spezifiziert sein. Die Anweisungen [DSZ] und [ISZ] beeinflussen nicht den Status des Carry- und des Out-Of-Range Flags.

Beispiel

Eine «Prüfsummen»-Routine dient zur Überprüfung der Unversehrtheit gespeicherter Datenwerte. Mit [#B] können Sie die Quersumme eines Bitmusters berechnen und mit der Quersumme des Musters zu einem späteren Zeitpunkt vergleichen.

Das folgende Programm bildet die Quersumme des Bitmusters eines gegebenen Datenregisters. Die Quersummen der Datenregister R_I bis R_A werden nacheinander ermittelt und in den Registern Y und Z zur doppelt langen Gesamtquersumme aufaddiert. Der Stackinhalt vor Zeile 012:



Die endgültige Prüfsumme wird in den Registern X und Y abgelegt.

Dieses Programm benutzt [DSZ], um einen Registerzeiger im Indexregister zu dekrementieren und um die bedingte Verzweigung zu steuern.

Tastenfolge	Anzeige			
[g] [P/R]	000–			
[f] CLEAR [PRGM]	000–			
[g] [LBL] D	001–43,22,		d	
[f] SET COMPL [UNSGN]	002–	42	3	Unsigned-Modus zur Bit-Addition.
4	003–		4	
[f] [WSIZE]	004–	42	44	Wortlänge 4.
[HEX]	005–		23	
A	006–		A	
[STO] [I]	007–	44	32	Speichert Adresse des höchsten Registers (R_A) in R_I.
0	008–		0	
[ENTER]	009–		36	Initialisiert Prüfsumme auf 0.
[g] [LBL] 0	010–43,22,		0	Start der Summationsschleife (Stack Lift freigegeben).
[RCL] [(i)]	011–	45	31	Rückruf des Inhalts des durch R_I indirekt adressierten Registers.

Tastenfolge	Anzeige			
g #B	012–	43	7	Summiert die Bits im X-Register.
+	013–		40	Addiert diese Summe zum niederwertigen Teil der momentanen Prüfsumme. Setzt ggf. den Carry-Flag.
x≷y	014–		34	Bringt höherwertigen Teil der Prüfsumme nach X.
0	015–		0	Setzt X auf 0.
g RLC	016–	43	C	Plaziert 1 in X, falls die vorangegangene Addition einen Übertrag ergab.
+	017–		40	Carry-Bit wird zum höherwertigen Teil der Prüfsumme addiert.
x≷y	018–		34	Niederwertiger Teil der Prüfsumme wird nach X zurückgeladen.
g DSZ	019–	43	23	Dekrementiert den Schleifenkontrollwert in R_I.
GTO 0	020–	22	0	Setzt Schleife fort, falls R_I noch nicht Null enthält.
g RTN	021–	43	21	

Berechnen Sie jetzt eine aktuelle Prüfsumme mit den folgenden 4-Bit Hexadezimalwerten in den Registern R_1 bis R_A.

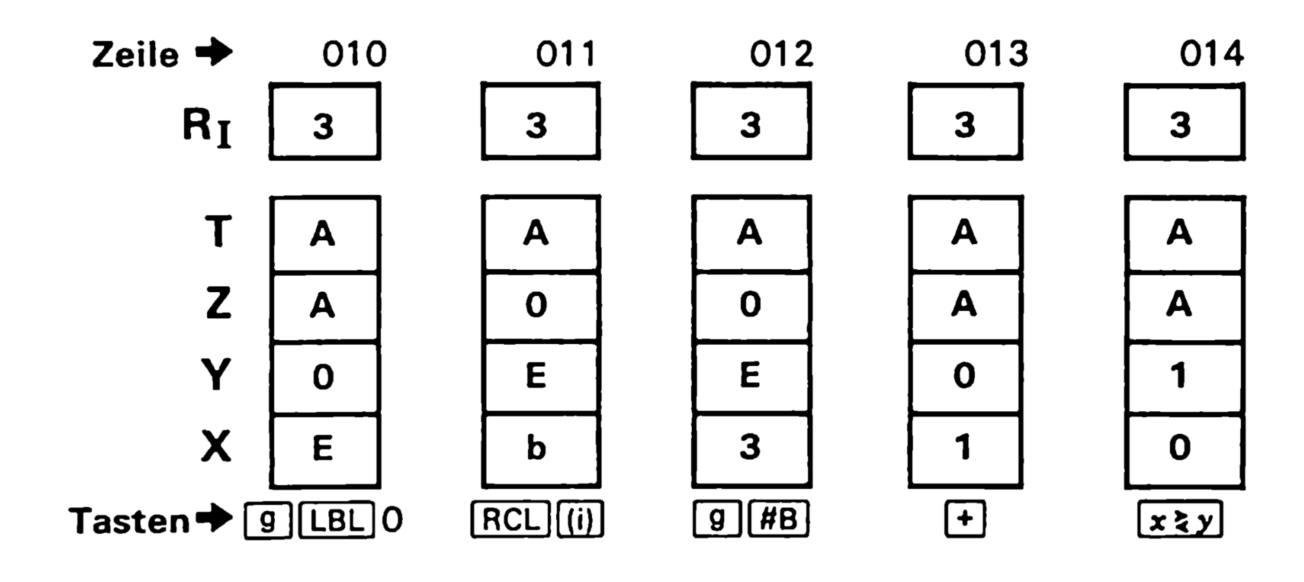
R_1: A	R_3: B	R_5: 3	R_7: A	R_9: D
R_2: 7	R_4: 1	R_6: D	R_8: 2	R_A: 6

Tastenfolge	Anzeige	(STATUS:0-04-0000)
g P/R		Schaltet in den Run-Modus zurück.
HEX		
A STO 1 ⋮ 6 STO A	A h ⋮ 6 h	Speichern Sie die gegebenen Werte in R_1 bis R_A.

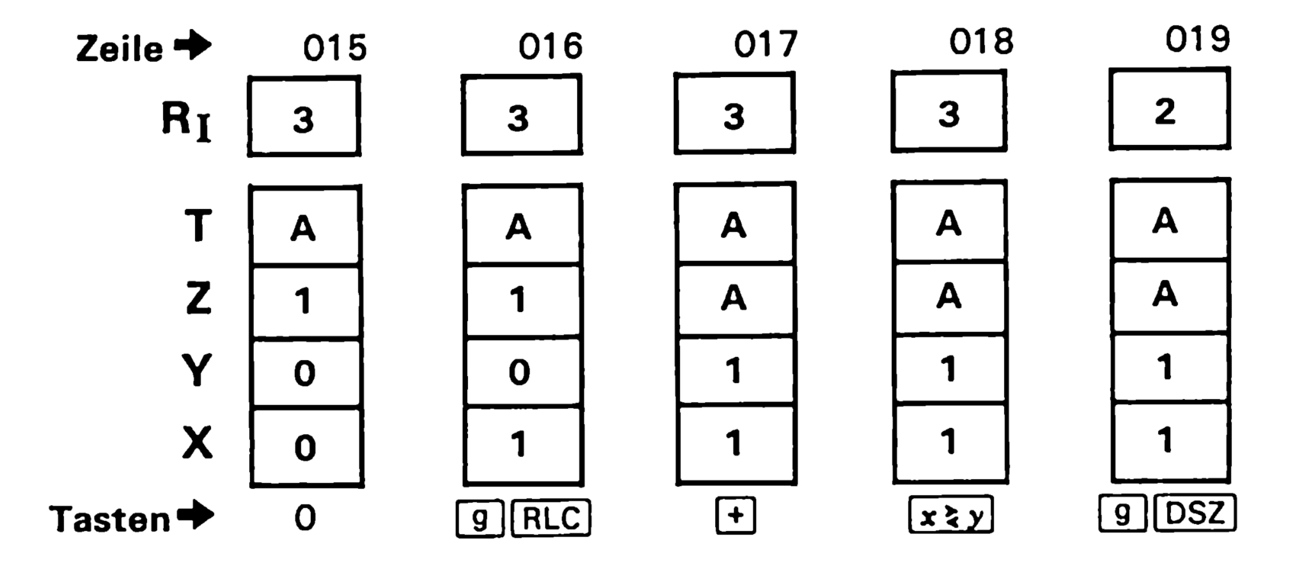
Tastenfolge	Anzeige	
[GSB] D	6 h	Niederwertige Bits der doppeltlangen Prüfsumme.
[x≷y]	1 h	Höherwertige Bits: die Prüfsumme unseres Bitmusters ist 16_{16} bzw. 22_{10}

Beim Schreiben oder Analysieren eines Programmes ist es oft nützlich, ein Diagramm der Stackinhalte vor und nach jeder Anweisung anzufertigen. Die folgenden Stackdiagramme zeigen die Bewegung der Stackinhalte im Schleifenabschnitt ([LBL] 0: Zeilen 010 bis 019) des Prüfsummenprogramms.

In der achten Iteration dieser Schleife wird in Zeile 013 das Carry-Bit gesetzt, wenn die Quersumme von R_3 zur bisherigen Prüfsumme (E_{16}) addiert wird, die damit die einfache Wortlänge überschreitet. Diese Iteration wird hier illustriert. (Das A in den Registern T und Z ist ein Überbleibsel aus den Zeilen 006 und 007.)



Zeile 012 bildet die Quersumme des von R_I adressierten Registers, und Zeile 013 addiert diese zum niederwertigen Teil der Prüfsumme. Die Zeilen 014 bis 017 addieren den Wert des Carry-Bits, der aus der vorhergehenden Addition stammt, zum höherwertigen Teil der Prüfsumme.

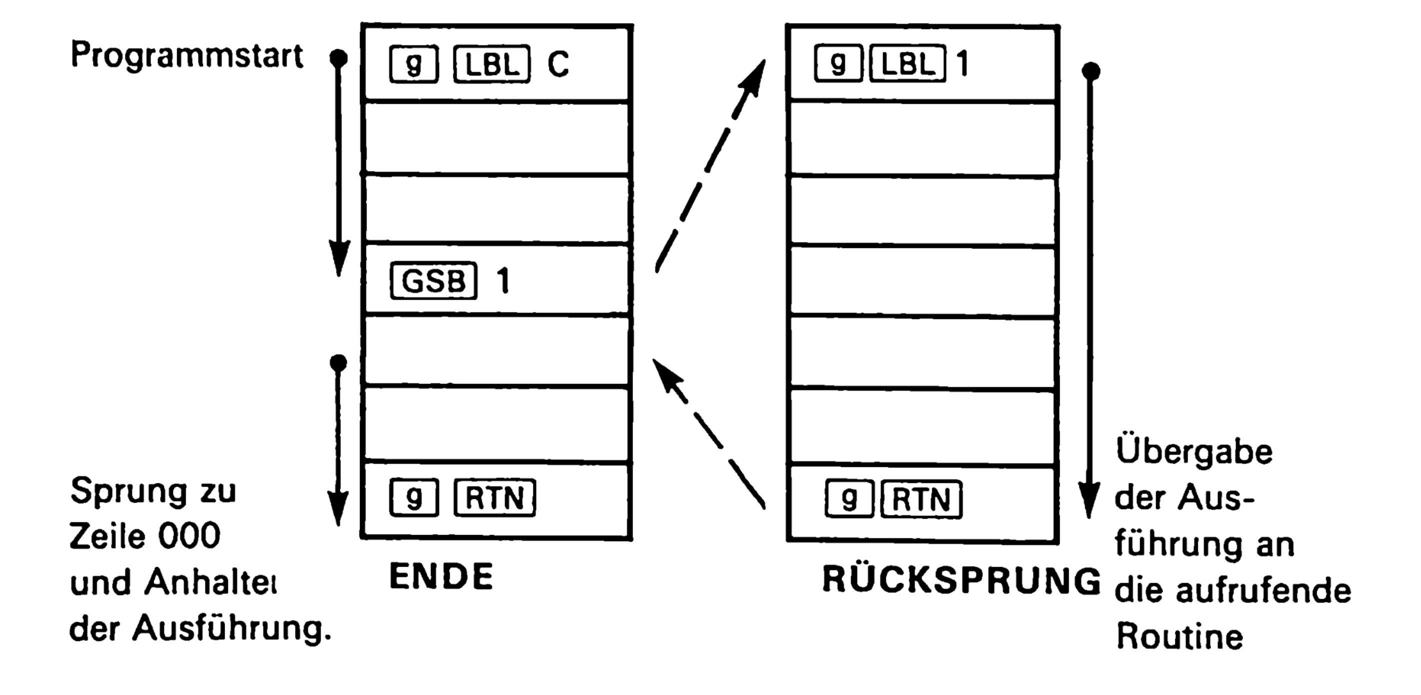


Zeile 019 dekrementiert die Adresse des Datenregisters; die nächste Schleife bildet dann die Quersumme eines anderen Registers.

Weitere Informationen

Unterprogramme

Ausführung. Die Anweisung [GSB] *Label* beinhaltet einfache Verzweigung mit besonderer Eigenschaft, daß eine «*anstehende Rücksprungbedingung*» gesetzt wird. Die Programmausführung wird mit dem spezifizierten Label fortgesetzt, bis die erste nachfolgende [RTN] Anweisung gefunden wird; die dann die Kontrolle an die auf die [GSB] Anweisung direkt folgende Zeile zurückgibt.



Verschachtelte Unterprogramme. Die «Verschachtelung» von Unterprogrammen – d.h. die Ausführung eines Unterprogramms innerhalb eines Unterprogramms – ist auf maximal vierfache Verschachtelungsebenen begrenzt (ohne die Hauptprogramm-Ebene).

Wenn Sie versuchen, ein mehr als vier Ebenen tief verschachteltes Unterprogramm aufzurufen, bricht der Rechner die Programmausführung ab und meldet **Error 5**, sobald er die [GSB] Anweisung auf der fünften Ebene findet. Die Anzahl der Unterprogramme mit weniger als fünf Verschachtelungsebenen ist lediglich durch die Speichergröße beschränkt.

Ausführung von [GSB] innerhalb eines Programms und über das Tastenfeld

Die Taste [GSB] wird für zwei verschiedene Funktionen benutzt: 1. um über Tastenfeld aus Programme aufzurufen, und 2. um innerhalb eines laufenden Programmes Unterprogramme aufzurufen. Beachten Sie, daß bei der Ausführung von [GSB] über das Tastenfeld keine anstehende Rücksprungbedingung gesetzt wird. Wenn im Programmlauf danach ein [RTN] angetroffen wird, stoppt die Programmausführung und kehrt zur Zeile 000 zurück – sofern die [RTN] Anweisung nicht zu einer nachfolgend programmierten [GSB] Anweisung gehört.

Anhang A

Fehlermeldungen und Flags

Fehlerbedingungen

Bei dem Versuch eine Operation mit einem unzulässigen Parameter auszuführen (etwa die Angabe eines nichtexistenten Flags), erscheint in der Anzeige die Meldung **Error** gefolgt von einer Kennzahl. Diese Fehlermeldung kann durch Drücken einer beliebigen Taste gelöscht werden. Nach der Löschung erscheint automatisch wieder der alte Inhalt der Anzeige. Im folgenden werden die einzelnen Fehlermeldungen in der Reihenfolge ihrer Kennzahlen aufgeführt.

Error 0: Unzulässige mathematische Operation

[÷], wo $x = 0$.
[RMD], wo $x = 0$.
[DBL÷], wo:

- $x = 0$.
- der resultierende *Quotient* größer als ein einzelnes Wort.

[DBLR], wo

- $x = 0$.
- der resultierende *Quotient* größer als 64 Bits.

[√x], wo $x < 0$.
[1/x], wo $x = 0$ (nur im dezimalen Gleitkomma-Modus).

Error 1: Unzulässiger Flag-, Fenster-, [FLOAT] oder [GTO] [·] Parameter

Angegebener Flag-Parameter ist größer als 5.
Angegebener Fenster-Parameter ist größer als 7.
Angegebener [FLOAT] Parameter ist größer als 9.
Angegebene [GTO] [·] Ziffer ist größer als 9.

Error 2: Unzulässige Anzahl von Bits

[MASKL] oder [MASKR], wo $|x|$ > momentane Wortgröße.
[RLn] oder [RRn], wo $|x|$ > momentane Wortgröße.
[RLCn] oder [RRCn], wo $|x|$ > momentane Wortgröße +1.
[SB], [CB] oder [B?], wo $|x| \geqslant$ momentane Wortgröße.

Error 3: Unzulässige Registeradresse

Das adressierte Speicherregister existiert nicht.

Error 4: Unzulässiges Label oder unzulässige Zeilennummer

Die durch das Label oder die Zeilennummer adressierte Programmzeile existiert nicht. Versuch mehr als 203 Programmzeilen zu laden.

Error 5: Unterprogrammverschachtelung zu tief

Eine Unterprogrammverschachtelung enthält mehr als vier Verschachtelungsebenen.

Error 6: Unzulässiger Registerinhalt

Versuch im dezimalen Gleitkomma-Modus ein Speicherregister (einschließlich R_I) aufzurufen, das keine Zahl im Gleitkomma-Format enthält.

Verwendung von [DSZ] oder [ISZ] im dezimalen Gleitkomma-Modus, wenn R_I keine Gleitkommazahl enthält.

Error 9: Service

Bei der Funktionsprüfung wurde ein Fehler entdeckt; beim Tastenfeldtest wurde eine falsche Taste gedrückt. Siehe Anhang C.

Pr Error (*Power Error*)

Löschung des Permanentspeichers wegen Stromausfall.

Flagverändernde Funktionen

Der Zustand zweier wichtiger Flags, des Carry Flags und der Out-Of-Range Flags, wird im *Integer Modus* durch bestimmte arithmetische Shift- Funktionen verändert. Die folgende Tabelle listet diese Funktionen:

× = gesetzt oder gelöscht 0 = immer gelöscht – = keine Auswirkung

Funktion	Auswirkung auf		Verwendete Register	
	Carry (4)	Out-Of-Range (5)	Operand(en)	Resultat
[+]	×	×*	X, Y	X
[-]	×	×*	X, Y	X
[×]	—	×*	X, Y	X
[÷]	×	×*	X, Y	X
[√x]	×	—	X	X
[DBL×]	—	0	X, Y	X, Y
[DBL÷]	×	0	X, Y, Z	X
[ABS]	—	×	X	X
[CHS]	—	×	X	X
[SL]	×	—	X	X
[SR]	×	—	X	X
[ASR]	×	—	X	X
[RL]	×	—	X	X
[RR]	×	—	X	X
[RLC]	×	—	X	X
[RRC]	×	—	X	X
[RLn]	×	—	X, Y	X
[RRn]	×	—	X, Y	X
[RLCn]	×	—	X, Y	X
[RRCn]	×	—	X, Y	X

* auch im dezimalen Gleitkomma-Modus

Anhang B

Operationstypen

Zifferneingaben beendende Operationen

Die meisten Operationen des Rechners, sei es bei Ausführung innerhalb eines Programms oder über das Tastenfeld, beenden die Eingabe von Ziffern. Dies besagt, daß der Rechner jede nach Abschluss einer dieser Operationen eingegebene Ziffer als Teil einer neuen Zahl auffasst.

Die folgenden *Zifferneingabe*-Operationen beenden die Zifferneingabe *nicht*:

[0] bis [9]	[·]	[CHS] im Gleitkomma-Modus.
[A] bis [F]	[EEX]	[BSP]

Operationen mit Auswirkung auf den Stack Lift

Die Operationen des Rechners lassen sich je nach ihrer Auswirkung auf den Stack Lift in drei Klassen aufteilen. Man unterscheidet stacksperrende (stack disabling), stackfreigebende (stack enabling) und neutrale Operationen.

Stacksperrende Operationen

Die Ausführung einer den Stack Lift sperrenden Operation bedingt, daß eine *nach* dieser Operation eingegebene Zahl den momentanen Inhalt des X-Registers überschreibt. Der Inhalt des Stacks wird nicht nach oben verschoben. Der HP-16C verfügt über die folgenden stacksperrenden Operationen:

[ENTER] [CLx]*

* Zusätzlich bewirkt auch [BSP] nach Abschluß einer Zifferneingabe ein Sperren des Stacks.

Neutrale Operationen

Neutrale Operationen sind dadurch gekennzeichnet, daß sie den momentanen Zustand (gesperrt oder freigegeben) des Stack Lifts nicht verändern.

Die folgenden Operationen sind neutral bezüglich des Stack Lifts.*

[HEX]	[WINDOW]	[MEM]	[PSE]
[DEC]	[<], [>]	[STATUS]	[R/S]
[OCT]	[GTO] [·] *nnn*	CLEAR [PREFIX]	[SST]
[BIN]	[P/R]	CLEAR [REG]	[BST]

SHOW {[HEX], [DEC], [OCT], [BIN]}

SET COMPL {[1's], [2's], [UNSGN]}

[FLOAT] (bei Ausführung im Gleitkomma-Modus)

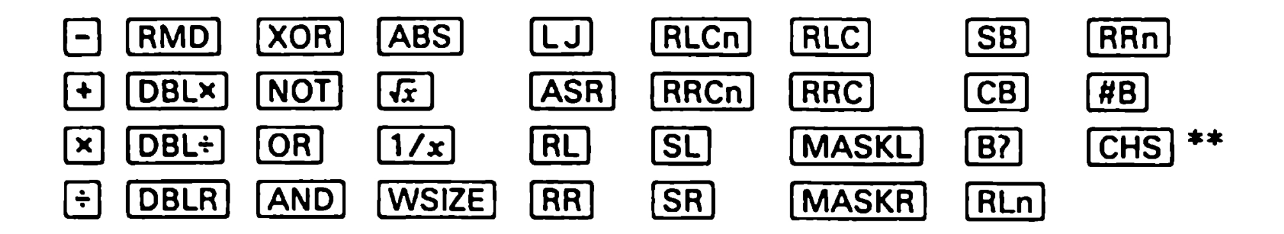
Stackfreigebende Operationen

Die meisten Rechneroperationen bewirken eine Freigabe des Stack Lifts. Jede *nach* einer dieser Operationen eingetastete Zahl bedingt ein Anheben des Stacks.

Alle bisher nicht als neutral oder stacksperrend klassifizierten Operationen sind automatisch stackfreigebend.

Speicheroperationen in das LAST X Register

Die folgenden Operationen retten den Inhalt des X-Registers in das LAST X-Register.



[−]	[RMD]	[XOR]	[ABS]	[LJ]	[RLCn]	[RLC]	[SB]	[RRn]
[+]	[DBL×]	[NOT]	[√x]	[ASR]	[RRCn]	[RRC]	[CB]	[#B]
[×]	[DBL÷]	[OR]	[1/x]	[RL]	[SL]	[MASKL]	[B?]	[CHS] **
[÷]	[DBLR]	[AND]	[WSIZE]	[RR]	[SR]	[MASKR]	[RLn]	

Bei der Ausführung der [FLOAT] Funktion im Integer Modus wird das LAST X-Register gelöscht.

---

* Zusätzlich sind alle Ziffereingabeoperationen vor Beendigung der Ziffereingabe neutral.

** Im Integer-Modus.

Fensterverschiebungen erhaltende Operationen

Die folgenden Operationen bewirken *kein* Zurücksetzen des Fensters, d.h. sie beinhalten keine [WINDOW] [0] Operation. Alle anderen Operationen bedingen die Anzeige der acht niederwertigen Stellen der Zahl im X-Register.

[<], [>]	[P/R]	[PSE]	[SST]
[ON]	[LBL]	[R/S]	[BST]
[STO]	[RTN]	[GSB]	[DSZ]
[MEM]	[SF], [CF], [F?]	[GTO]	[ISZ]
[STATUS]			

SHOW {(momentane Basis)} CLEAR {[PRGM], [REG], [PREFIX]}

[x≤y], [x<0], [x>y], [x>0], [x≠y], [x≠0][x=y], [x=0]

Vorwahltasten

Der HP-16C verfügt über die folgenden Vorwahltasten:

[f]	[STO]	[SF]	[LBL]	[GTO][·]
[g]	[RCL]	[CF]	[GSB]	
[FLOAT]	[WINDOW]	[F?]	[GTO]	

Im dezimalen Gleitkomma-Modus inaktive Operationen

[SL]	[RR]	[MASKL]	[SB]	[NOT]
[LJ]	[RRC]	[MASKR]	[CB]	[OR]
[SR]	[RLn]	[RMD]	[B?]	[<]
[ASR]	[RLCn]	[DBLR]	[WSIZE]	[>]
[RL]	[RRn]	[DBL÷]	[XOR]	[A] to [F]
[RLC]	[RRCn]	[DBL×]	[AND]	

SHOW {[HEX], [DEC], [OCT], [BIN]}

Im Integer-Modus sind nur die Operationen [·], [EEX] und [1/x] nicht aktiv.

Anhang C

Batterien, Gewährleistung und Serviceinformation

Batterien

Die Stromversorgung des HP-16C erfolgt über einen Satz von drei Batterien. Bei «normalem» Gebrauch des Rechners wird mit einem Satz Alkalibatterien eine ungefähre Betriebsdauer von 3 Monaten erreicht. Der HP-16C wird mit Alkalibatterien ausgeliefert; Sie können jedoch auch Silberoxidbatterien (die die Betriebsdauer zumindest verdoppeln sollten) verwenden.

Drei frische Alkalibatterien liefern eine reine Programmrechenzeit von mindestens 80 Stunden (hierbei sei angemerkt, daß die Ausführung von Programmen die Form des Rechnerbetriebs mit dem höchsten Stromverbrauch ist).* Bei der Verwendung von Silberoxidbatterien erhöht sich die reine Programmrechenzeit auf mindestens 180 Stunden. Diese Angaben beziehen sich, wie bereits erwähnt, auf die aufwendigste Betriebsart, nämlich die Ausführung von Programmen; bei allen anderen Betriebsarten ist der Stromverbrauch weitaus geringer. Wenn nur die Anzeige eingeschaltet ist, d.h. es laufen keine Programme ab und es werden keine Tasten gedrückt, ist der Stromverbrauch minimal.

Bei ausgeschaltetem Rechner bleibt der Inhalt des Permanentspeichers über einen Zeitraum erhalten, der dem der Haltbarkeit der Batterien außerhalb des Rechners entspricht – mindestens anderthalb Jahre bei Alkalibatterien bzw. mindestens zwei Jahre bei Silberoxidbatterien.

Die aktuelle Lebenszeit der Batterien hängt davon ab, wie oft Sie den Rechner benutzen, ob Sie ihn mehr zur Ausführung von Programmen oder für manuelle Berechnungen verwenden, und mit welchen Funktionen Sie arbeiten.*

Sowohl die zusammen mit dem Rechner gelieferten Batterien als auch die im folgenden empfohlenen Ersatzbatterien können *nicht* nachgeladen werden.

---

* Der Stromverbrauch des HP-16C wird wesentlich durch die jeweilige Betriebsart bestimmt: ausgeschaltet (bei Erhaltung des Permanentspeichers); betriebsbereit (nur die Anzeige ist eingeschaltet); rechnend (ein Programm läuft ab, eine Berechnung wird durchgeführt, oder eine Taste ist gedrückt). Solange der Rechner eingeschaltet ist, besteht die typische Auslastung aus einer Mischung der beiden Betriebsarten «betriebsbereit» und «rechnend». Die aktuelle Lebensdauer der Batterien hängt daher sehr stark davon ab, wie lange der Rechner in jeder der drei Betriebsarten benutzt wird.

WARNUNG

Versuchen Sie nicht die Batterien nachzuladen, werfen Sie sie nicht in offenes Feuer und lagern Sie sie nicht in der Nähe großer Wärmeeinstrahlung. Die Batterien können in diesen Fällen auslaufen bzw. explodieren.

Die folgenden Batterien werden als Ersatzbatterien für Ihren HP-16C empfohlen (nicht alle Batterientypen sind in allen Ländern erhältlich):

Alkalibatterien	**Silberoxidbatterien**
Eveready A76*	Eveready 357*
UCAR A76	UCAR 357
RAY-O-VAC RW82	RAY-O-VAC RS76 oder RW42
National oder Panasonic LR44	Duracell MS76 oder 10L14
Varta 4276	Varta 541

Anzeige abfallender Batteriespannung

Bei nachlassender Batteriespannung leuchtet in der linken unteren Ecke der Anzeige ein Stern (⋆) auf.

Bei Verwendung von Alkalibatterien:

- Der Rechner kann nach dem ersten Aufleuchten des Sterns noch mindestens zwei Stunden zur reinen Programmausführung verwendet werden.**
- Wenn der Rechner nach dem ersten Aufleuchten des Sterns ausgeschaltet wird, bleibt der Inhalt des Permanentspeichers noch wenigstens für einen Monat erhalten.

Bei Verwendung von Silberoxidbatterien:

- Der Rechner kann nach dem ersten Aufleuchten des Sterns noch mindestens 15 Minuten zur reinen Programmausführung verwendet werden.**
- Wenn der Rechner nach dem ersten Aufleuchten des Sterns ausgeschaltet wird, bleibt der Inhalt des Permanentspeichers noch mindestens für eine Woche erhalten.

* Nicht erhältlich in Großbritannien und der Republik Irland.

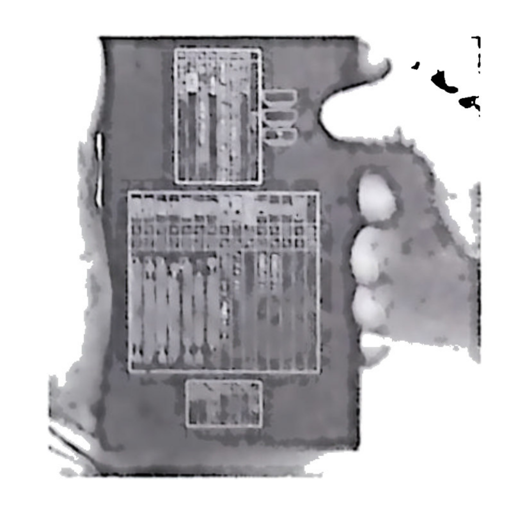
** Die angegebene Zeit bezieht sich auf den Fall, daß laufend Programme ausgeführt werden, d.h. auf die Betriebsart auf Seite 102). Wird der Rechner für manuelle Berechnungen benutzt – eine Mischung der Betriebsarten «betriebsbereit» und «rechnend» – so kann der Rechner noch wesentlich länger nach dem ersten Auftauchen des Sterns verwendet werden.

Einsetzen neuer Batterien

Der Inhalt des Permanentspeichers bleibt nach dem Ausbau der alten Batterien noch für einige Sekunden erhalten (sofern Sie den Rechner vor dem Entfernen der Batterien ausgeschaltet haben). Sie haben daher normalerweise genügend Zeit zum Austausch der Batterien zur Verfügung, ohne dabei Daten oder Programme zu verlieren. Der Inhalt des Permanentspeichers geht erst verloren, wenn über längere Zeit keine Batterien im Rechner sind.

Um neue Batterien einzubauen, verfahren Sie wie folgt:

1. Vergewissern Sie sich, daß der Rechner ausgeschaltet ist.
2. Halten Sie den Rechner wie in der Abbildung gezeigt und drücken Sie den Batteriefachdeckel nach außen, bis er sich leicht öffnet.



3. Greifen Sie den Batteriefachdeckel an der Außenkante und ziehen Sie ihn vom Rechner ab.



Hinweis: Achten Sie während der nächsten beiden Schritte darauf, daß keine Tasten gedrückt werden, solange sich keine Batterien im Rechner befinden. In diesem Fall kann der Inhalt des Permanentspeichers gelöscht werden oder aber die Tastenfeldkontrolle verloren gehen (d.h. der Rechner reagiert nicht mehr auf das Drücken von Tasten).



4. Drehen Sie den Rechner herum und schütteln Sie ihn leicht, bis die Batterien in Ihre Hand fallen.

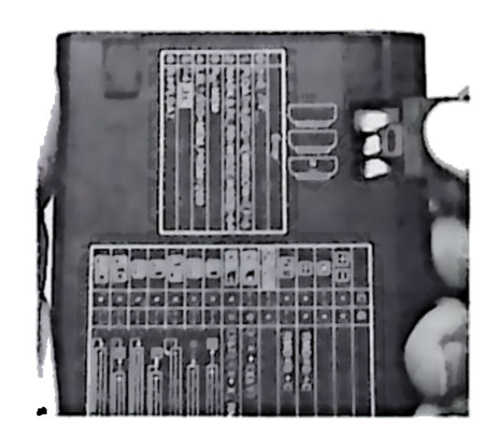
VORSICHT

Erneuern Sie im nächsten Schritt *alle drei* Batterien. Wenn Sie nur eine oder zwei Batterien austauschen, besteht die Gefahr, daß eine der alten Batterien ausläuft. Achten Sie des weiteren darauf, daß Sie die Batterien nicht verkehrt herum einsetzen. In diesem Fall kann der Inhalt des Permanentspeichers verloren gehen oder die betreffende Batterie beschädigt werden.

5. Halten Sie die beiden Plastikstreifen im Batteriefach nach oben und setzen Sie drei neue Batterien ein. Die Batterien müssen so eingelegt werden, daß die flachen (mit + markierten) Seiten zum nächsten Gummifuß zeigen. (Siehe Illustration auf dem Rechnergehäuse.)

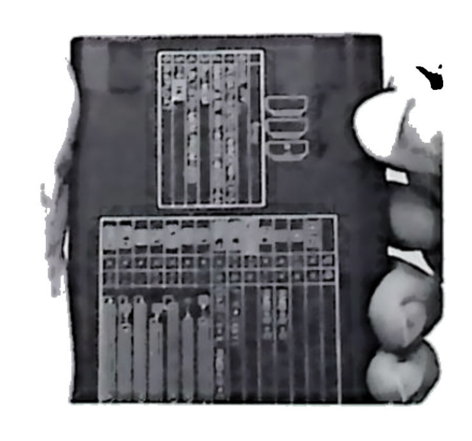


6. Schieben Sie die Halterung des Batteriefachdeckels in die Aussparung im Rechnergehäuse.



7. Drücken Sie den Batteriefachdeckel nach unten, bis er mit dem Rechnergehäuse abschließt, und pressen Sie ihn anschließend nach innen, bis er leicht einrastet.

8. Schalten Sie den Rechner ein. Wenn der Permanentspeicher aus irgendeinem Grund gelöscht wurde, erscheint in der Anzeige die Meldung **Pr Error**. Sie können diese Meldung dann durch Drücken einer beliebigen Taste aus der Anzeige löschen.



Funktionsprüfung

Wenn sich der Rechner nicht einschalten läßt oder anderweitig nicht korrekt zu arbeiten scheint, sollten Sie den Rechner mit Hilfe der folgenden Schritte austesten:

Falls der Rechner *nicht* auf Tastendruck reagiert:

1. Drücken Sie gleichzeitig die Tasten [D] und [ON] und lassen Sie sie wieder los. Da bei dieser Operation der Inhalt des X-Registers verändert wird, sollten Sie dieses Register nach der Funktionsprüfung löschen.
2. Wenn der Rechner danach immer noch nicht auf das Drücken von Tasten reagiert, nehmen Sie die Batterien heraus und setzen Sie wieder ein. Achten Sie darauf, daß die Batterien korrekt in das Batteriefach eingesetzt sind.
3. Falls der Rechner auch weiterhin nicht auf Tastendruck reagiert, belassen Sie die Batterien im Batteriefach und schließen Sie die Kontakte im Batteriefach kurz. (Biegen Sie die Plastikstreifen zurück, um die Kontakte an den Seiten des Batteriefachs freizulegen.) *Sie brauchen die Kontakte nur für einen Moment kurzzuschließen.* Bei dieser Operation geht der Inhalt des Permanentspeichers verloren, und Sie müssen zum Wiedereinschalten des Rechners die [ON]-Taste unter Umständen mehrmals drücken.
4. Setzen Sie frische Batterien ein, wenn sich der Rechner jetzt immer noch nicht einschalten läßt. Wenn der Rechner auch danach nicht auf Tastendruck reagiert, ist er defekt.

Falls der Rechner auf Tastendruck reagiert:

1. Halten Sie bei ausgeschaltetem Rechner die Taste [ON] gedrückt und drücken Sie zusätzlich die Taste [×].
2. Lassen Sie zunächst die [ON]-Taste und danach die [×]-Taste los. Diese Tastenfolge löst eine vollständige Überprüfung der elektronischen Schaltkreise des Rechners aus. Bei korrekter Arbeitsweise des Rechners sollte nach einer Zeitspanne von 15 Sekunden (in der die Meldung **running** in der Anzeige blinkt), die Anzeige **-8,8,8,8,8,8,8,8,8,8** erscheinen. Zusätzlich sollten sämtliche Statusanzeigen (mit Ausnahme der Spannungsabfall-Anzeige ★) eingeschaltet sein.* Wenn die Anzeige leer bleibt, die Meldung **Error 9** oder sonst in irgendeiner Form nicht das gewünschte Ergebnis zeigt, deutet dies auf einen Gerätefehler hin.**

* Einige der Statusanzeigen, die am Ende dieses Tests eingeschaltet sind, werden vom HP-16C normalerweise nicht verwendet.

** Wenn der Rechner als Ergebnis des [ON]/[×] Tests oder des [ON]/[+] Tests die Meldung **Error 9** anzeigt, und Sie den Rechner trotzdem weiter verwenden wollen, empfiehlt es sich, zuvor den Permanentspeicher (wie auf Seite 20 beschrieben) zu löschen.

Hinweis: Überprüfungen der Rechnerelektronik werden auch durch die Tastenkombinationen [ON]/[+] und [ON]/[÷] ausgelöst.* Diese Tests dienen der Funktionskontrolle des Rechners während der Herstellung und eventueller Reparaturen.

Wenn sich Ihr Verdacht auf Fehlfunktion des Rechners aufgrund einer korrekten Anzeige in Schritt 3 nicht bestätigt hat, so besteht die Möglichkeit, daß Sie bei der Bedienung des Rechners einen Fehler gemacht haben. In diesem Fall empfehlen wir, den auf Ihr spezielles Problem zutreffenden Abschnitt dieses Handbuchs noch einmal durchzuarbeiten. Wenn Sie danach immer noch Schwierigkeiten feststellen, sollten Sie sich mit Hewlett-Packard unter einer der im Abschnitt «Service» (siehe Seite 110) gelisteten Adressen in Verbindung setzen.

---

* Durch die Tastenkombination [ON]/[+] wird eine dem zuvor beschriebenen Test ähnliche Überprüfung ausgelöst, die jedoch nicht von selbst abbricht. Dieser Test wird innerhalb von 15 Sekunden nach Drücken einer beliebigen Taste beendet. Die Kombination [ON]/[÷] löst eine Überprüfung des Tastenfelds und der Anzeige aus. Nach dem Loslassen der [ON]-Taste leuchten zunächst bestimmte Segmente der Anzeige auf. Zur Fortsetzung des Tests müssen alle Tasten des Tastenfelds gedrückt werden, und zwar von links nach rechts, beginnend mit der ersten Reihe. Beim Drücken der einzelnen Tasten leuchten jeweils verschiedene Segmente der Anzeige auf. Wenn der Rechner korrekt arbeitet und *alle Tasten in der richtigen Reihenfolge gedrückt werden*, wird nach dem Drücken der letzten Taste die Zahl **16** angezeigt. (Die Taste [ENTER] sollte sowohl zusammen mit den Tasten der dritten Reihe als auch zusammen mit den Tasten der vierten Reihe gedrückt werden.) Wenn der Rechner nicht fehlerfrei arbeitet oder *eine Taste außerhalb der korrekten Reihenfolge gedrückt wurde*, erscheint in der Anzeige die Meldung **Error 9**. *Beachten Sie dabei, daß der Rechner nicht notwendig reparaturbedürftig ist, wenn diese Fehlermeldung durch das Drücken einer Taste in der falschen Reihenfolge ausgelöst wurde.* Dieser Test kann durch Drücken einer beliebigen Taste außerhalb der Reihenfolge (was natürlich eine **Error 9** Meldung induziert) vorzeitig beendet wird. Sowohl die Anzeige **Error 9** als die Anzeige **16** kann durch Drücken einer beliebigen Taste gelöscht werden.

Gewährleistung

Hewlett-Packard gewährleistet, daß das Produkt frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn das Produkt – direkt oder über einen autorisierten Hewlett-Packard Vertragshändler – an eine Hewlett-Packard Serviceniederlassung eingeschickt wird. Die Gewährleistungsfrist beträgt 12 Monate ab Verkaufsdatum.

Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden. Schäden, die auf unsachgemäße Veränderungen des Produkts durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt.

Die Gewährleistung gilt nur bei Vorlage der vollständig ausgefüllten Service-Karte in Verbindung mit entweder

a) dem von einem autorisierten HP-Vertragshändler ausgestellten und unterschriebenen Original-Kaufbeleg aus dem das Kaufdatum und die Seriennummer des Produkts hervorgehen

   oder

b) der Originalrechnung von Hewlett-Packard.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Regelung unberührt.

Änderungsverpflichtung

Die Produkte von Hewlett-Packard werden auf der Basis der zum Zeitpunkt der Herstellung gegebenen technischen Spezifikationen verkauft. Hewlett-Packard übernimmt keine Verpflichtung zur nachträglichen Anpassung oder Modifikation einmal verkaufter Produkte.

Gewährleistungsinformation

Wenn Sie Fragen zu dieser Gewährleistungserklärung haben, nehmen Sie bitte Kontakt mit einem autorisierten Hewlett-Packard Händler oder einer Hewlett-Packard Verkaufs- oder Serviceniederlassung auf. Sollte dies nicht möglich sein, wenden Sie sich bitte an:

- In Europa:

Hewlett-Packard S.A.
7, rue du Bois-du-Lan
P.O. Box
CH-1217 Meyrin 2
Genf
Schweiz

Hinweis: Bitte senden Sie an diese Adresse *keine* Geräte zur Reparatur.

- In den Vereinigten Staaten:

Hewlett-Packard
Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330
Telefon: (503) 758-1010

- In allen anderen Ländern:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, California 94304
U.S.A.
Telefon: (415) 857-1501

Hinweis: Bitte senden Sie an diese Adresse *keine* Geräte zur Reparatur.

Service

Hewlett-Packard unterhält Serviceniederlassungen in vielen Ländern der Welt. Diese Niederlassungen stehen Ihnen jederzeit für eine eventuelle Reparatur zur Verfügung, auch wenn die Gewährleistungsfrist von einem Jahr bereits abgelaufen sein sollte. Reparaturen nach Ablauf der Gewährleistungsfrist sind kostenpflichtig.

Normalerweise erfolgt die Reparatur und der Rückversand von Hewlett-Packard Produkten innerhalb von 5 Werktagen. In Abhängigkeit von der Auslastung der Serviceniederlassung kann diese Zeitspanne im Einzelfall überschritten werden.

Service-Zentrale in den Vereinigten Staaten

In den Vereinigten Staaten erreichen Sie die Service-Zentrale von Hewlett-Packard für Taschenrechner- und Taschencomputer-Produkte unter der folgenden Anschrift:

Hewlett-Packard Company
Corvallis Division Service Department
P.O. Box 999/1000 N.E. Circle Blvd.
Corvallis, Oregon 97330, U.S.A.
Telefon: (503) 757-2000

Serviceniederlassungen in Europa

Hewlett-Packard unterhält Serviceniederlassungen in den folgenden Ländern. Wenn das Land, in dem Sie sich befinden, nicht aufgeführt ist, sollten Sie sich mit dem HP-Händler, bei dem Sie Ihr Gerät erworben haben, in Verbindung setzen.

BELGIEN
HEWLETT-PACKARD BELGIUM SA/NV
Boulevard de la Woluwe 100
Woluwelaan
B-1200 BRÜSSEL
Tel.: (2) 762 32 00

DÄNEMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 BIRKEROD (Kopenhagen)
Tel.: (02) 81 66 40

DEUTSCHLAND
HEWLETT-PACKARD GmbH
Vertriebszentrale
Berner Strasse 117
Postfach 560 140
D-6000 FRANKFURT 56
Tel.: (0611) 5004-1

FINNLAND
HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 ESPOO 10 (Helsinki)
Tel.: (90) 455 02 11

FRANKREICH
HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F-91947 LES ULIS CEDEX
Tel.: (6) 907 78 25

GROSSBRITANNIEN
HEWLETT-PACKARD Ltd.
King Street Lane
Winnersh, Wokingham
GB BERKSHIRE RG11 5AR
Tel.: (734) 78 47 74

NIEDERLANDE
HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
NL-1181 KK AMSTELVEEN (Amsterdam)
P.O. Box 67
Tel.: (020) 47 20 21

ITALIEN
HEWLETT-PACKARD S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 CERNUSCO SUL NAVIGLIO (Milan)
Tel.: (2) 90 36 91

NORWEGEN
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 OESTERAAS (Oslo)
Tel.: (2) 17 11 80

ÖSTERREICH
HEWLETT-PACKARD GmbH
Wagramerstrasse-Lieblgasse
A-1220 WIEN
Tel.: (0222) 23 65 11

OSTEUROPA
Wenden Sie sich an die unter Österreich angegebene Adresse.

SCHWEDEN
HEWLETT-PACKARD SVERIGE AB
Skalholtsgatan 9
Box 19
S-163 93 SPÅNGA
Tel.: (0046) 8 750 20 00

SCHWEIZ
HEWLETT-PACKARD (SCHWEIZ) AG
Allmend 2
CH-8967 WIDEN
Tel.: (057) 31 21 11

SPANIEN
HEWLETT-PACKARD ESPANOLA S.A.
Calle Jerez 3
E MADRID 16
Tel.: (1) 458 2600

Internationale Serviceinformation

Nicht jede Hewlett-Packard Serviceniederlassung bietet Service für alle Hewlett-Packard Produkte. Wenn Sie jedoch Ihr Gerät bei einem autorisierten Hewlett-Packard Händler gekauft haben, können Sie sicher sein, daß in dem Land des Erwerbs auch Servicemöglichkeiten bestehen.

Wenn Sie sich nicht in dem Land befinden, in dem Sie Ihr Gerät erworben haben, befragen Sie die lokale Hewlett-Packard Serviceniederlassung nach den dort verfügbaren Servicemöglichkeiten. Wenn kein Service verfügbar ist, senden Sie Ihr Gerät an die zuvor aufgeführte Adresse der Service-Zentrale in den Vereinigten Staaten. Unter der gleichen Adresse können Sie auch eine Liste der Serviceniederlassungen anderer Länder anfordern.

Sämtliche der durch den Versand entstehenden Kosten gehen zu Ihren Lasten.

Reparaturkosten

Bei Reparaturen außerhalb der Gewährleistungsfrist werden Standardsätze zugrunde gelegt, die den Arbeitslohn und den Materialaufwand beinhalten. In den Vereinigten Staaten unterliegt der gesamte Rechnungsbetrag der lokalen Umsatzsteuer des Kunden. In europäischen Ländern ist der Rechnungsbetrag mehrwertsteuerpflichtig bzw. unterliegt ähnlichen Steuern. Der jeweilige Steueranteil wird in der Rechnung getrennt ausgewiesen.

Für Produkte, die durch Gewalteinwirkung oder sonstigen Mißbrauch beschädigt worden sind, gelten diese festen Reparatursätze nicht. In diesen Fällen wird die Reparatur individuell nach Arbeitszeit und Materialaufwand berechnet.

Service-Garantie

Bei Reparaturen außerhalb der Gewährleistungsfrist wird eine Garantie auf das Material und die Arbeitsleistung von 90 Tagen gegeben. Diese Garantiefrist gilt ab dem Reparaturdatum.

Versandanweisungen

Wenn Sie Ihr defektes Gerät einsenden, fügen Sie bitte bei:

- Eine vollständig ausgefüllte Service-Karte, einschließlich einer Fehlerbeschreibung.
- Die Originalrechnung oder einen sonstigen Kaufnachweis, sofern die Reparatur in die einjährige Gewährleistungsfrist fällt.

Zur Vermeidung von Transportschäden sollte das Gerät (zusammen mit der Service-Karte, einer kurzen Beschreibung des Fehlers, sowie, falls erforderlich, dem Kaufnachweis) nur in der Originalverpackung oder einer adäquaten Schutzverpackung versandt werden. Transportschäden fallen nicht unter die einjährige Gewährleistung. Das verpackte Gerät sollte entweder direkt oder über Ihren HP-Vertragshändler zur nächsten Hewlett-Packard Serviceniederlassung geschickt werden. (Wenn Sie sich nicht in dem Land befinden, in dem Sie Ihr Gerät erworben haben, beziehen Sie sich bitte auf den Abschnitt «Internationale Serviceinformation».) Hewlett-Packard empfiehlt Ihnen, den Transport gegebenenfalls versichern zu lassen.

Die Versandkosten zur Hewlett-Packard Serviceniederlassung gehen zu Ihren Lasten, unabhängig davon, ob sich das Gerät noch in der Gewährleistungsfrist befindet oder nicht.

Bei Reparaturen innerhalb der Gewährleistungsfrist übernimmt die Serviceniederlassung die Kosten für den Rückversand. Bei Reparaturen ausserhalb dieser Frist sind die Rücksendungskosten im Rechnungsbetrag enthalten.

Sonstiges

Hewlett-Packard bietet keine Service-Verträge an. Entwurf und Ausführung des Produkts und der Elektronik sind geistiges Eigentum von Hewlett-Packard; Service-Handbücher können daher nicht an Kunden abgegeben werden.

Sollten Sie weitere servicebezogene Fragen haben, setzen Sie sich bitte mit Ihrem HP-Vertragshändler oder der nächstgelegenen Hewlett-Packard Serviceniederlassung in Verbindung.

Benutzerberatung

Sollten beim Einsatz Ihres Gerätes in bestimmten Anwendungsfällen Fragen auftauchen, so wenden Sie sich an den HP-authorisierten Fachhändler, bei dem Sie das Gerät bezogen haben.
Bezugsquellennachweis über den Fachhandel erhalten Sie in der Bundesrepublik Deutschland über:

HEWLETT-PACKARD GmbH
Vertriebszentrale/Werbeabteilung
Bernerstraße 117
Postfach 560140
D-6000 FRANKFURT/M 56

Temperaturspezifikationen

- Betriebstemperatur: 0° bis 55° C
- Lagertemperatur: –40° bis 65° C

Anhang D

Programme zur Formatumwandlung

Das zur Darstellung einer Zahl verwendete Format ist in der Regel vom Typ des jeweiligen Rechners abhängig. Daher ist es bei der Übertragung von Daten zwischen Rechnern verschiedenen Typs oft notwendig, Zahlen von einem Format in ein anderes umzuwandeln. Dieser Anhang enthält zwei Programme zur Umwandlung von Zahlen zwischen dem vorgeschlagenen IEEE Standard für ein binäres Gleitkomma-Format und dem vom HP-16C verwendeten dezimalen Gleitkomma-Format.*

Formate

Der vorgeschlagene IEEE Standard für einfach genaues, binäres Gleitkomma-Format hat die folgende Gestalt.

s	*e*	*f*
31	30 — 23	22 — 0

Dieses 32-Bit Format umfaßt: 1 Vorzeichenbit *s*
8 Bit zur Darstellung des Exponenten *e*
23 Bit zur Darstellung der Mantisse

Der Wert *v* einer Zahl *x* im X-Register wird wie folgt interpretiert:

a) Falls $e=255$ und $f\neq 0$ ist $v=\text{NaN}$ (*not a number* – keine Zahl).

b) Falls $e=255$ und $f=0$ ist $v=(-1)^s\infty$.

c) Falls $0<e<255$ ist $v=(-1)^s\,2^{(e-127)}\,(1.f)$.

d) Falls $e=0$ und $f\neq 0$ ist $v=(-1)^s\,2^{(-126)}\,(0.f)$.

e) Falls $e=0$ und $f=0$ ist $v=(-1)^s\,0$.

Im dezimalen Gleitkomma-Modus des HP-16C werden die folgenden Konventionen benutzt:

* Der Standard für das binäre Gleitkomma-Format ist ein Vorschlag des IEEE Computer Society's Floating-Point Committee (Task 754) und wurde veröffentlicht in *Computer*, März 1981, Seiten 51 bis 62.

IEEE Zahl	X-Register	Carry (Flag 4)	Out-of-Range (Flag 5)
0	0	0	0
−0	0	1	0
±∞	$\pm 9.999999999 \times 10^{99}$	1	1
Andere Zahlen	Wie unter c) und d) definiert	0	0
Keine Zahl	$(-1)^s\,(0.f)\,2^{23}$	1	0

Umwandlung vom IEEE Format in das HP-16C Format

Das folgende Programm wandelt eine Zahl vom einfach genauen, binären IEEE Gleitkomma-Format in das dezimale Gleitkomma-Format um.

TASTENFOLGEN	ANZEIGE	TASTENFOLGEN	ANZEIGE
[g] [LBL] B	001–43,22, b	[x⇄y]	018– 34
[HEX]	002– 23	8	019– 8
[f] SET COMPL [2's]	003– 42 2	[f] [MASKL]	020– 42 7
2	004– 2	[g] [x=y]	021– 43 49
0	005– 0	[GTO] 4	022– 22 4
[f] [WSIZE]	006– 42 44	[R↓]	023– 33
[f] [SL]	007– 42 A	[g] [x=0]	024– 43 40
[ENTER]	008– 36	[GTO] 3	025– 22 3
[ENTER]	009– 36	[x⇄y]	026– 34
[g] [x=0]	010– 43 40	1	027– 1
[GTO] 2	011– 22 2	8	028– 8
1	012– 1	[f] [SB]	029– 42 4
8	013– 8	[g] [LBL] 1	030–43,22, 1
[f] [MASKR]	014– 42 8	[g] [F?] 4	031–43, 6, 4
[f] [AND]	015– 42 20	[CHS]	032– 49
[f] [XOR]	016– 42 10	[x⇄y]	033– 34
[g] [LSTx]	017– 43 36	8	034– 8

TASTENFOLGEN	ANZEIGE	TASTENFOLGEN	ANZEIGE
[f] [RLn]	035- 42 E	[g] [CLx]	051- 43 35
9	036- 9	[g] [x≠y]	052- 43 0
7	037- 7	[GTO] 5	053- 22 5
[-]	038- 30	1	054- 1
[g] [CF] 4	039-43, 5, 4	4	055- 4
[g] [LBL] 2	040-43,22, 2	5	056- 5
[f] [FLOAT] [·]	041-42,45,48	[ENTER]	057- 36
[g] [RTN]	042- 43 21	[g] [LBL] 5	058-43,22, 5
[g] [LBL] 3	043-43,22, 3	[x⇄y]	059- 34
1	044- 1	[g] [F?] 4	060-43, 6, 4
8	045- 8	[CHS]	061- 49
[f] [SB]	046- 42 4	[g] [ASR]	062- 43 b
[x⇄y]	047- 34	[x⇄y]	063- 34
[GTO] 1	048- 22 1	[g] [SF] 4	064-43, 4, 4
[g] [LBL] 4	049-43,22, 4	[GTO] 2	065- 22 2
[R↓]	050- 33		

Beispiele: ([STATUS]:2-32-0000)

Tastenfolgen	Anzeige	
[HEX] 80000000	80000000 h	-0.
[GSB] B	0.000000 00	C gesetzt.
[HEX] 7F800000		$+\infty$.
[GSB] B	9.999999 99	C und G gesetzt.
[HEX] 00800000		$2^{-126} \times (1.00 \ldots 00)$.
[GSB] B	1.175494-38	
[HEX] 3F800001		$2^{0} \times (1.00 \ldots 01) = 1 + 2^{-23}$.
[GSB] B	1.000000 00	
[f] CLEAR [PREFIX]	1000000119	

Umwandlung vom HP-16C Format in das IEEE Format

Das folgende Programm wandelt eine Zahl vom dezimalen Gleitkomma-Format des HP-16C in das einfach genaue, binäre IEEE Gleitkomma-Format um. Das Programm setzt Flag 5 (Out-Of-Range), wenn das Ergebnis der Umwandlung ± ∞ ist. (Die in diesem Programm verwendeten Labels kommen im letzten Programm nicht vor, so daß Programme gleichzeitig im Speicher gehalten werden können.)

TASTENFOLGEN	ANZEIGE	TASTENFOLGEN	ANZEIGE
[g] [LBL] A	001–43,22, A	0	025– 0
[f] SET COMPL [2's]	002– 42 2	[f] [WSIZE]	026– 42 44
[HEX]	003– 23	8	027– 8
[g] [CF] 4	004–43, 5, 4	0	028– 0
[g] [CF] 5	005–43, 5, 5	[+]	029– 40
[g] [x=y]	006– 43 49	1	030– 1
[g] [RTN]	007– 43 21	8	031– 8
9	008– 9	[f] [MASKL]	032– 42 7
D	009– d	[f] [AND]	033– 42 20
[+]	010– 40	[g] [F?] 4	034–43, 6, 4
[x≷y]	011– 34	[g] [ISZ]	035– 43 24
[g] [CF] 0	012–43, 5, 0	[f] [SL]	036– 42 A
[g] [x<0]	013– 43 2	[RCL] I	037– 45 32
[g] [SF] 0	014–43, 4, 0	F	038– F
[g] [ABS]	015– 43 8	F	039– F
[x≷y]	016– 34	[g] [x>y]	040– 43 3
[g] [x<0]	017– 43 2	[GTO] 7	041– 22 7
[GTO] 9	018– 22 9	[x≷y]	042– 34
1	019– 1	[R↓]	043– 33
[+]	020– 40	[R↓]	044– 33
[g] [LBL] 6	021–43,22, 6	[g] [CLx]	045– 43 35
[STO] I	022– 44 32	[g] [R↑]	046– 43 33
[R↓]	023– 33	[g] [R↑]	047– 43 33
2	024– 2	[g] [SF] 5	048–43, 4, 5

TASTENFOLGEN	ANZEIGE	TASTENFOLGEN	ANZEIGE
[g][LBL] 7	049–43,22, 7	[g][LBL] 9	062–43,22, 9
[R↓]	050– 33	[g][ABS]	063– 43 8
[f][OR]	051– 42 40	3	064– 3
[g][F?] 0	052–43, 6, 0	0	065– 0
[GSB] 8	053– 21 8	[g][x≤y]	066– 43 1
9	054– 9	[x⇄y]	067– 34
[f][RRn]	055– 42 F	[R↓]	068– 33
[g][CF] 4	056–43, 5, 4	0	069– 0
[g][RTN]	057– 43 21	[x⇄y]	070– 34
[g][LBL] 8	058–43,22, 8	[f][SB]	071– 42 4
8	059– 8	[÷]	072– 10
[f][SB]	060– 42 4	0	073– 0
[g][RTN]	061– 43 21	[GTO] 6	074– 22 6

Beispiele: ([STATUS]:2-32-0000)

Tastenfolge	Anzeige	
[f][FLOAT][·]		
8 [f][EEX] 72	8 72	
[GSB] A	7F800000 h	G gesetzt. Überlauf auf +∞.
[f][FLOAT][·]		
1.404 [f][EEX]	1.404 00	
45 [CHS]	1.404 –45	
[GSB] A	1 h	
[f][FLOAT][·]		
3.141592654	3.141592654	π.
[GSB] A	40490Fdb h	

Funktionsindex

Dieser Abschnitt enthält eine Kurzbeschreibung aller Rechnerfunktionen mit Seitenangabe der ausführlichen Darstellung. Die Funktionen sind in folgenden Gruppen zusammengefasst:

[ON] schaltet die Anzeige ein und aus **(Seite 16)**. [ON]/[-] löscht den Permanentspeicher **(Seite 20)**; [ON]/[·] ändert die Ziffern- und Dezimaltrennzeichen **(Seite 61)**; weitere Kombinationen bewirken Rechnertest **(Seite 106)**.

Löschfunktionen

[BSP] Backspace. Im Run-Modus: löscht letzte Ziffer, bzw. die gesamte Anzeige nach Abschluß der Zifferneingabe **(Seite 17)**. Im Programm-Modus: löscht die derzeitige Anweisung **(Seite 83)**.

[CLx] Clear X. Löscht den Inhalt des X-Registers auf 0 **(Seite 17)**.

CLEAR [PRGM] Clear Program Memory. Im Run-Modus: Programmpointer wird auf Zeile 000 gesetzt, *keine* Löschung von Programmzeilen. Im Programm-Modus: Löschung des gesamten Programmspeichers **(Seite 73)**.

CLEAR [REG] Clear Registers. Löscht alle Datenspeicher-Register **(Seite 68)**.

CLEAR [PREFIX] Löscht Vorwahl-Eingabe. Anulliert alle Vorwahlen einer unvollständig eingegebenen Tastenfolge **(Seite 17)**. Zeigt kurzzeitig die volle 10stellige Mantisse der Zahl im X-Register (nur im Gleitkomma-Modus) **(Seite 58)**.

Zifferneingabe

Zifferntasten [0] bis [9] [A] bis [F]. Können nur

im entsprechenden Zahlenbasis-Modus benutzt werden **(Seite 28).**

[.] Dezimalpunkt. Nur im Gleitkommamodus **(Seite 58).**

[ENTER] Kopiert die Zahl im X-Register in das Y-Register, beendet Ziffereingabe und sperrt den Stack. Dient zur Trennung von Zahleneingaben **(Seite 22).**

[CHS] Vorzeichenwechsel. Bildet das entsprechende Komplement oder den Negativwert der Zahl im X-Register **(Seiten 30 und 58).**

[EEX] Eingabe des Exponenten. Nur im Gleitkommamodus. Nach [EEX] eingegebene Ziffern werden als Exponenten zur Basis 10 betrachtet **(Seite 58).**

Stackmanipulation

[x⇄y] Austausch X und Y. Vertauscht die Inhalte der Register X und Y **(Seite 23).**

[R↓], [R↑] Roll down, Roll up. Rollt die Inhalte des Stack um ein Register nach unten oder nach oben **(Seite 23).**

Zahlen- und Anzeigesteuerung

[HEX], [DEC], [OCT], [BIN] Zahlenbasis-Modi. Setzen die Anzeige im Integer-Modus auf das gewählte Zahlensystem **(Seite 28).**

SHOW {[HEX], [DEC], [OCT], [BIN]}. Kurzzeitige Anzeige des X-Registerinhalts in der spezifizierten Zahlenbasis **(Seite 29).**

SET COMPL {[1's], [2's], [UNSGN]} Komplement-Modus. Setzt Einerkomplement-, Zweierkomplement oder Unsigned-Modus **(Seite 30).**

[WSIZE] Wortlänge. Der Absolutwert der Zahl im X-Register (0 bis 64) bestimmt die Wortlänge, danach Stack Drop **(Seite 32).**

[WINDOW] {0 bis 7}. Zeigt das angegebene 8stellige Segment der Zahl im X-Register an **(Seite 33).**

[<], [>] Verschieben des Anzeige-Fensters um eine Stelle nach rechts oder links. **(Seite 33).**

[SF], [CF] Setzen und Löschen von Flags. Setzt bzw. löscht den angegebenen Flag (0 bis 5) **(Seite 36).**

[STATUS] Zeigt kurzzeitig den derzeitigen Komplement-Modus, die Wortlänge und den Flagstatus an **(Seite 37).**

[FLOAT] {0 bis 9, [.]}. Setzt dezimalen Gleitkomma-Modus; die angegebene Anzahl Dezimalstellen wird angezeigt; [.] setzt wissenschaftliche Notation. Beim Umschalten vom Integer- in den Gleitkomma-Modus werden die Werte der Register X und Y als Wert (y) (2^x) im X-Register gespeichert, die andern Stackregister werden gelöscht **(Seite 56).**

Mathematische Funktionen

[+], [−], [×], [÷]. Arithmetische Operatoren; Stack fällt nach der Ausführung. Im Integer-Modus wird nach [÷] nur der ganzzahlige Anteil angezeigt **(Seite 41).**

[RMD] Rest. Berechnet $|y| \text{MOD} |x|$ Vorzeichen von y wird übernommen; der Stack fällt **(Seite 43).**

[√x], [1/x] Quadratwurzel und Kehrwert. Operiert auf dem X-Register; kein Stack Drop. [1/x] nur im Gleitkommamodus. Im Integer-Modus wird nur der ganzzahlige Teil von [√x] angezeigt **(Seiten 44 und 61).**

[DBLx], [DBL÷], [DBLR] Doppeltgenaue Multiplikation und Division, doppeltgenauer Rest. [DBLx] ergibt Produkt doppelter Wortlänge in den Registern X und Y, [DBL÷] und [DBLR] bearbeiten einen Dividenden doppelter Wortlänge in Y und Z (Dividend in X) und legen das Ergebnis in X ab **(Seite 52).**

[ABS] Absolutwert. Bildet Betrag der Zahl im X-Register **(Seite 44).**

Bitmanipulationen

[SL], [SR] Shift links, Shift rechts. Verschiebt die Bits im X-Register um eine Stelle nach rechts oder links. Das herausgeschobene Bit geht in das Carry-Bit, ein neues Bit ist immer Null **(Seite 46).**

[ASR] Arithmetischer Shift nach rechts. Verschiebt alle Bits in X um eine Stelle nach rechts und dupliziert das Vorzeichenbit links (nur im Einer- und im Zweierkomplement-Modus) **(Seite 47).**

[RL], [RR] Rotate links, Rotate rechts. Rotiert die Bits im X-Register um eine Stelle nach rechts oder links; ein am einen Ende das Wort verlassendes Bit wird ans andere Ende und in Carry transportiert **(Seite 48).**

[RLC], [RRC] Rotate nach links/rechts durch Carry. Wie oben, außer daß die herausgeschobenen Bits durch das Carry-Bit wandern, bevor sie ans andere Wortende gebracht werden **(Seite 48).**

[RLn], [RRn], [RLCn], [RRCn] Rundumverschiebung nach links/rechts (durch das Carry-Bit oder nicht) des im Y-Register stehenden Bitmusters um die in X stehende Anzahl Stellen, danach fällt der Stack, das neue Bitmuster steht in X **(Seite 49).**

[LJ] linksbündige Ausrichtung. Richtet das Wort im X-Register linksbündig aus, plaziert es im Y-Register. Im X-Register steht die Anzahl Stellen, um die verschoben wurde **(Seite 47).**

[MASKL], [MASKR] Maske links/rechts. Erzeugt eine links- oder rechtsbündige Maske gesetzter Bits. Die Bitanzahl wird durch den Betrag der Zahl im X-Register bestimmt **(Seite 50).**

[SB], [CB] Setze/lösche Bit. Setzt ein Bit, dessen Position durch den Wert in X bestimmt ist, in einem in Y stehenden Bitmuster auf 1 oder löscht es auf 0. Die Bits sind von rechts nach links mit 0 bis (Wortlänge −1) numeriert **(Seite 50).**

[#B] Anzahl von Bits. Summiert die Bits im X-Register und legt das Ergebnis in X ab. Keine Stackbewegung **(Seite 52).**

[NOT], [OR], [AND], [XOR] Logische Operatoren. [NOT] invertiert das Bitmuster im X-Register. Die anderen drei Funktionen operieren auf den Werten in X und Y

und legen das Ergebnis in X ab (der Stack fällt). **(Seite 44).**

Speicher

[STO] Kopiert den Inhalt des X-Registers in das angegebene Datenregister (0 bis .F, [I], [(i)]) **(Seiten 66, 69).**

[RCL] Kopiert das angegebene Datenregister in das X-Register **(Seiten 66, 69).**

[x↔I], [x↔(i)] Direkter und indirekter Austausch mit dem Indexregister, siehe Indexregisterfunktionen.

[LSTx] Last X Register. Ruft den Anzeigewert vor der letzten Operation ins X-Register zurück **(Seite 23).**

[MEM] Speicheraufteilung. Zeigt kurzzeitig an: 1) die Zahl noch zur Verfügung stehender Programmzeilen vor der Umwandlung weiterer Datenregister, und 2) die Anzahl derzeit verfügbarer Datenregister **(Seite 65).**

CLEAR [REG], CLEAR [PRGM] Löschen aller Datenregister/des Programmspeichers. Siehe Löschfunktionen.

Indexregisterfunktionen

[I] Indexregister (R_I). Speicherregister, das auch zur Programmsteuerung (mit [GTO] und [GSB]), und Schleifensteuerung (mit [DSZ] und [ISZ]) benutzt werden kann **(Seite 68).**

[(i)] Indirekte Operationen. Wird zur indirekten Adressierung von Datenregistern benutzt, und ist die einzige Möglichkeit, auf Register oberhalb $R_{.F}$ zuzugreifen. **(Seite 69).**

[x↔I] Austausch von X und R_I. Vertauscht den Wert im X-Register mit dem im Indexregister **(Seite 69).**

[x↔(i)] Vertauscht den Wert im X-Register mit dem Inhalt des von R_I indirekt adressierten Datenregisters **(Seite 69)**

[DSZ] [ISZ] Schleifensteuerung über R_I. Siehe Abfragen und Verzweigungen **(Seite 124).**

Programmierung

[P/R] Programm-/Run-Modus. Setzt den Rechner auf Programm-Modus, um Programmzeilen abzuspeichern, oder auf Run-Modus, um Programme oder andere Operationen auszuführen **(Seite 72).**

[LBL] {0 bis F} Label. Dient zur Adressierung von Programmen **(Seite 73).**

[RTN] Return. Stoppt die Ausführung eines laufenden Programms und setzt den Rechner auf Zeile 000. In einem Unterprogramm wird die Programmausführung mit der Zeile nach der zugehörigen [GSB] Anweisung fortgesetzt **(Seite 74).**

[R/S] Run/Stop. Startet oder stoppt die Ausführung mit der derzeitigen Zeile im Programmspeicher **(Seite 76).**

[PSE] Pause. Unterbricht den Programmlauf kurzzeitig, um das X-Register anzuzeigen **(Seite 75).**

[GTO] *label.* Programmausführung wird mit dem angegebenen Label fortgesetzt. Programmierbar **(Seite 87).**

[GTO] [·] *nnn*. Positioniert den Rechner auf die existierende Zeile *nnn*. Nicht programmierbar **(Seite 82)**.

[GSB] *Label*. Aufruf eines Unterprogramms. Bei Ausführung innerhalb eines Programms wird die Kontrolle an das spezifizierte Unterprogramm übergeben; die Kontrolle wird durch die nächste [RTN] Anweisung an das Hauptprogramm zurückgegeben. Bei Ausführung über das Tastenfeld wird mit [GSB] ein mit dem Label gekennzeichnetes Programm gestartet **(Seite 87)**.

[SST] Einzelschritt. Programm-Modus: Rechner wird zeilenweise im Programmspeicher vorwärtsbewegt; bei gehaltener Taste fortlaufend. Run-Modus: zeigt die derzeitige Programmzeile an und führt die darin gespeicherte Anweisung aus. Danach Sprung zur nächsten Anweisung **(Seite 82)**.

[BST] Einzelschritt zurück. Bewegt Rechner um eine Programmzeile zurück (im Programm-Modus auch fortlaufend). Zeigt Zeilennummer und Inhalt der voranstehenden Programmzeile **(Seite 82)**.

Vergleichsabfragen

[F?], [B?] Flag/Bit gesetzt? Fragt den spezifizierten Flag, das spezifizierte Bit ab. Falls gesetzt, wird die Programmausführung fortgesetzt, andernfalls wird zuvor eine Programmzeile übersprungen **(Seite 89)**.

[x≤y], [x<0], [x>y], [x>0], [x≠y], [x≠0], [x=y], [x=0] Bei jeder Abfrage wird der Wert im X-Register mit Null oder mit dem Inhalt des Y-Registers verglichen. Ist die Vergleichsbedingung erfüllt, wird die Programmausführung fortgesetzt; wenn nicht, wird zuvor eine Programmzeile übersprungen **(Seite 88)**.

[DSZ], [ISZ] Dekrement/Inkrement und überspringen falls Null. Dekrementiert bzw. inkrementiert den Wert im Indexregister und überspringt die nächste Programmzeile, falls der neue Indexwert Null ist **(Seite 90)**.

Sachindex

Seitenzahlen in **Fettdruck** stehen für Hauptreferenzen; Seitenzahlen in Normaldruck für zusätzliche Referenzen.

E

F

G

H

I

K

M

N

O

P

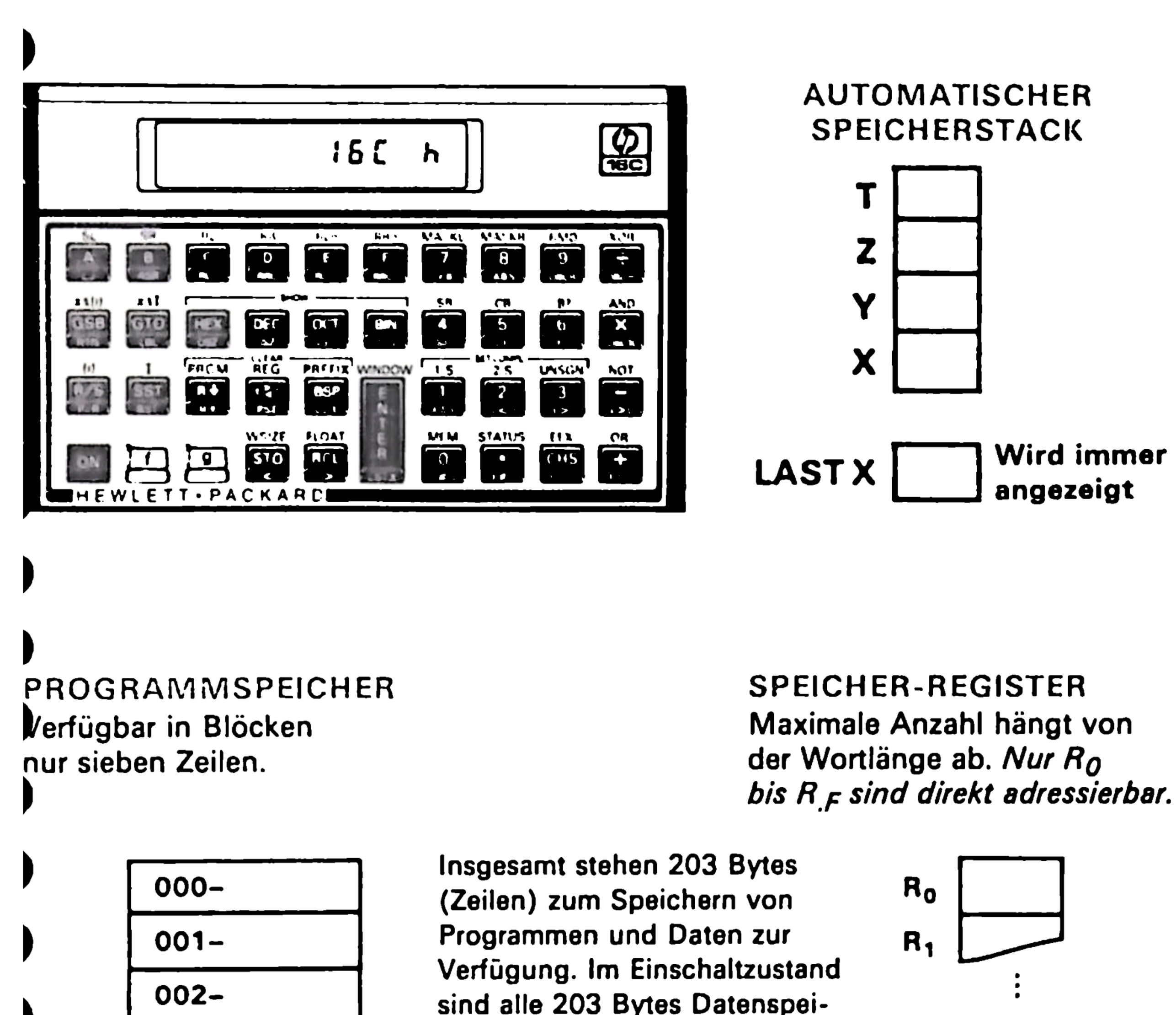
T

U

V



Tastenfeld und Speicher des HP-16C



PROGRAMMSPEICHER
Verfügbar in Blöcken nur sieben Zeilen.

SPEICHER-REGISTER
Maximale Anzahl hängt von der Wortlänge ab. *Nur R_0 bis $R_{.F}$ sind direkt adressierbar.*

Insgesamt stehen 203 Bytes (Zeilen) zum Speichern von Programmen und Daten zur Verfügung. Im Einschaltzustand sind alle 203 Bytes Datenspeicher-Registern zugewiesen, die bei Bedarf in Programmspeicher umgewandelt werden.

Beim Eintasten jeder siebten Programmzeile (001, 008, 015, ..., 197) werden sieben weitere Bytes an Datenspeicher in Programmspeicher umgewandelt. Die Anzahl der Register, die für diese sieben Bytes umgewandelt werden müssen, hängt von der jeweiligen Wortlänge ab.

Indexregister (kann nicht umgewandelt werden).

VERKAUFSNIEDERLASSUNGEN:

Hewlett-Packard GmbH:
6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 5004-1
7030 Böblingen, Herrenbergerstraße 110, Tel. (07031) 667-1
4000 Düsseldorf 11, Emanuel-Leutze-Straße 1 (Seestern), Tel. (0211) 5971-1
2000 Hamburg 60, Kapstadtring 5, Tel. (040) 63804-1
8028 Taufkirchen, Eschenstraße 5, Tel. (089) 6117-1
3000 Hannover 91, Am Großmarkt 6, Tel. (0511) 466001
8500 Nürnberg, Neumeyerstraße 90, Tel. (0911) 522083/87
1000 Berlin 30, Keithstraße 2-4, Tel. (030) 249086
6800 Mannheim, Roßlauer Weg 2-4, Tel. (0621) 70050
7910 Neu-Ulm, Messerschmittstraße 7, Tel. (0731) 70241
7517 Waldbronn 2, Hewlett-Packard Straße, Tel. (07243) 602-1

Hewlett-Packard (Schweiz) AG:
Allmend 2, CH-8967 Widen, Tel. (057) 31 21 11

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:
Wagramerstraße-Lieblgasse 1, A-1220 Wien, Tel. (0222) 236511

Hewlett-Packard S.A., Europa-Zentrale:
7, rue du Bois-du-Lan, Postfach, CH-1217 Meyrin 2-Genf, Schweiz

SERVICENIEDERLASSUNGEN:

Hewlett-Packard GmbH:
6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 5004-1

Hewlett-Packard (Schweiz) AG:
Allmend 2, CH-8967 Widen, Tel. (057) 31 21 11

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:
Wagramerstraße-Lieblgasse 1, A-1220 Wien, Tel. (0222) 236511



00016-90002 German - 3/86

Printed in Canada